

## TRAITE DE COOPERATION EN MATIERE DE BREVETS

PCT

NOTIFICATION DE LA RECEPTION DE  
L'EXEMPLAIRE ORIGINAL

(règle 24.2.a) du PCT)

Expéditeur: le BUREAU INTERNATIONAL

Destinataire:

DENIS, Hervé  
 Bull S.A.  
 68, route de Versailles 28 107.1999  
 PC58D20  
 F-78434 Louveciennes Cedex  
 FRANCE

Date d'expédition (jour/mois/année) 18 novembre 1999 (18.11.99)	NOTIFICATION IMPORTANTE
Référence du dossier du déposant ou du mandataire PCT 3682/HD	Demande internationale no PCT/FR99/02629

Il est notifié au déposant que le Bureau international a reçu l'exemplaire original de la demande internationale précisée ci-après.

Nom(s) du ou des déposants et de l'Etat ou des Etats pour lesquels ils sont déposants:

BULL S.A. (pour tous les Etats désignés sauf US)  
 NACHEF, Armand etc. (pour US seulement)

Date du dépôt international : 28 octobre 1999 (28.10.99)  
 Date(s) de priorité revendiquée(s) : 30 octobre 1998 (30.10.98).  
 Date de réception de l'exemplaire original  
 par le Bureau international : 12 novembre 1999 (12.11.99)  
 Liste des offices désignés :

EP : AT,BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE  
 National : US

## ATTENTION

Le déposant doit soigneusement vérifier les indications figurant dans la présente notification. En cas de divergence entre ces indications et celles que contient la demande internationale, il doit aviser immédiatement le Bureau international.

En outre, l'attention du déposant est appelée sur les renseignements donnés dans l'annexe en ce qui concerne

- ☒ les délais dans lesquels doit être abordée la phase nationale  
☒ la confirmation des désignations faites par mesure de précaution  
☐ les exigences relatives aux documents de priorité.

Une copie de la présente notification est envoyée à l'office récepteur et à l'administration chargée de la recherche internationale.

Bureau international de l'OMPI 34, chemin des Colombettes 1211 Genève 20, Suisse	Fonctionnaire autorisé Yolaine CUSSAC
n° de télécopieur (41-22) 740.14.35	n° de téléphone (41-22) 338.83.38

**THIS PAGE BLANK (USPTO)**

## RENSEIGNEMENTS CONCERNANT LES DELAIS DANS LESQUELS DOIT ETRE ABORDEE LA PHASE NATIONALE

Il est rappelé au déposant qu'il doit aborder la "phase nationale" auprès de chacun des offices désignés indiqués sur la notification de la réception de l'exemplaire original (formulaire PCT/IB/301) en payant les taxes nationales et en remettant les traductions, telles qu'elles sont prescrites par les législations nationales.

Le délai d'accomplissement de ces actes de procédure est de **20 MOIS** à compter de la date de priorité ou, pour les Etats désignés qui ont été élus par le déposant dans une demande d'examen préliminaire international ou dans une élection ultérieure, de **30 MOIS** à compter de la date de priorité, à condition que cette élection ait été effectuée avant l'expiration du 19<sup>e</sup> mois à compter de la date de priorité. Certains offices désignés (ou élus) ont fixé des délais qui expirent au-delà de 20 ou 30 mois à compter de la date de priorité. D'autres offices accordent une prolongation des délais ou un délai de grâce, dans certains cas moyennant le paiement d'une taxe supplémentaire.

En plus de ces actes de procédure, le déposant devra dans certains cas satisfaire à d'autres exigences particulières applicables dans certains offices. Il appartient au déposant de veiller à remplir en temps voulu les conditions requises pour l'ouverture de la phase nationale. La majorité des offices désignés n'envoient pas de rappel à l'approche de la date limite pour aborder la phase nationale.

Des informations détaillées concernant les actes de procédure à accomplir pour aborder la phase nationale auprès de chaque office désigné, les délais applicables et la possibilité d'obtenir une prolongation des délais ou un délai de grâce et toutes autres conditions applicables figurent dans le volume II du Guide du déposant du PCT. Les exigences concernant le dépôt d'une demande d'examen préliminaire international sont exposées dans le chapitre IX du volume I du Guide du déposant du PCT.

GR et ES sont devenues liées par le chapitre II du PCT le 7 septembre 1996 et le 6 septembre 1997, respectivement, et peuvent donc être élues dans une demande d'examen préliminaire international ou dans une élection ultérieure présentée le 7 septembre 1996 (ou à une date postérieure) ou le 6 septembre 1997 (ou à une date postérieure), respectivement, quelle que soit la date de dépôt de la demande internationale (voir le second paragraphe, ci-dessus).

Veuillez noter que seul un déposant qui est ressortissant d'un Etat contractant du PCT lié par le chapitre II ou qui y a son domicile peut présenter une demande d'examen préliminaire international.

## CONFIRMATION DES DESIGNATIONS FAITES PAR MESURE DE PRECAUTION

Seules les désignations expresses faites dans la requête conformément à la règle 4.9.a) figurent dans la présente notification. Il est important de vérifier si ces désignations ont été faites correctement. Des erreurs dans les désignations peuvent être corrigées lorsque des désignations ont été faites par mesure de précaution en vertu de la règle 4.9.b). Toute désignation ainsi faite peut être confirmée conformément aux dispositions de la règle 4.9.c) avant l'expiration d'un délai de 15 mois à compter de la date de priorité. En l'absence de confirmation, une désignation faite par mesure de précaution sera considérée comme retirée par le déposant. Il ne sera adressé aucun rappel ni invitation. Pour confirmer une désignation, il faut déposer une déclaration précisant l'Etat désigné concerné (avec l'indication de la forme de protection ou de traitement souhaitée) et payer les taxes de désignation et de confirmation. La confirmation doit parvenir à l'office récepteur dans le délai de 15 mois.

## EXIGENCES RELATIVES AUX DOCUMENTS DE PRIORITE

Pour les déposants qui n'ont pas encore satisfait aux exigences relatives aux documents de priorité, il est rappelé ce qui suit.

Lorsque la priorité d'une demande nationale, régionale ou internationale antérieure est revendiquée, le déposant doit présenter une copie de cette demande antérieure, certifiée conforme par l'administration auprès de laquelle elle a été déposée ("document de priorité"), à l'office récepteur (qui la transmettra au Bureau international) ou directement au Bureau international, avant l'expiration d'un délai de 16 mois à compter de la date de priorité, étant entendu que tout document de priorité peut être présenté au Bureau international avant la date de publication de la demande internationale, auquel cas ce document sera réputé avoir été reçu par le Bureau international le dernier jour du délai de 16 mois (règle 17.1.a)).

Lorsque le document de priorité est délivré par l'office récepteur, le déposant peut, au lieu de présenter ce document, demander à l'office récepteur de le préparer et de le transmettre au Bureau international. La requête à cet effet doit être formulée avant l'expiration du délai de 16 mois et peut être soumise au paiement d'une taxe (règle 17.1.b)).

Si le document de priorité en question n'est pas fourni au Bureau international, ou si la demande adressée à l'office récepteur de préparer et de transmettre le document de priorité n'a pas été faite (et la taxe correspondante acquittée, le cas échéant) avant l'expiration du délai applicable mentionné aux paragraphes précédents, tout Etat désigné peut ne pas tenir compte de la revendication de priorité; toutefois, aucun office désigné ne peut décider de ne pas tenir compte de la revendication de priorité avant d'avoir donné au déposant la possibilité de remettre le document de priorité dans un délai raisonnable en l'espèce.

Lorsque plusieurs priorités sont revendiquées, la date de priorité à prendre en considération aux fins du calcul du délai de 16 mois est la date du dépôt de la demande la plus ancienne dont la priorité est revendiquée.

**THIS PAGE BLANK (USPTO)**

# PCT

## REQUÊTE

Le soussigné requiert que la présente demande internationale soit traitée conformément au Traité de coopération en matière de brevets.

Réservé à l'office récepteur

Demande internationale n°

Date du dépôt international

Nom de l'office récepteur et "Demande internationale PCT"

Référence du dossier du déposant ou du mandataire (facultatif)  
(12 caractères au maximum)

PCT 3682/HD

Cadre n° I TITRE DE L'INVENTION

**Procédé de commande d'une fonction exécutable par des commandes spécifiques à des logiciels différents.**

Cadre n° II DÉPOSANT

Nom et adresse : (Nom de famille suivi du prénom: pour une personne morale, désignation officielle complète. L'adresse doit comprendre le code postal et le nom du pays. Le pays de l'adresse indiquée dans ce cadre est l'Etat où le déposant a son domicile si aucun domicile n'est indiqué ci-dessous.)

BULL S.A.

68, route de Versailles  
78434 LOUVECIENNES CEDEX  
FRANCE

☐ Cette personne est aussi inventeur.

n° de téléphone  
(33) 1 39.66.61.80

n° de télécopieur  
(33) 1 39.66.61.73

n° de téléimprimeur

Nationalité (nom de l'Etat) : FRANCE

Domicile (nom de l'Etat) : FRANCE

Cette personne est déposant pour : ☐ tous les États désignés ☒ tous les États désignés sauf les États-Unis d'Amérique ☐ les États-Unis d'Amérique seulement ☐ les États indiqués dans le cadre supplémentaire

Cadre n° III AUTRE(S) DÉPOSANT(S) OU (AUTRE(S)) INVENTEUR(S)

Nom et adresse : (Nom de famille suivi du prénom: pour une personne morale, désignation officielle complète. L'adresse doit comprendre le code postal et le nom du pays. Le pays de l'adresse indiquée dans ce cadre est l'Etat où le déposant a son domicile si aucun domicile n'est indiqué ci-dessous.)

Nachef Armand

6 Place Georges Pompidou  
78180 MONTIGNY LE BRETONNEUX  
FRANCE

Cette personne est :

☐ déposant seulement

☒ déposant et inventeur

☐ inventeur seulement  
(Si cette case est cochée, ne pas remplir la suite.)

Nationalité (nom de l'Etat) : FRANCE

Domicile (nom de l'Etat) : FRANCE

Cette personne est déposant pour : ☐ tous les États désignés ☐ tous les États désignés sauf les États-Unis d'Amérique ☒ les États-Unis d'Amérique seulement ☐ les États indiqués dans le cadre supplémentaire

☐ D'autres déposants ou inventeurs sont indiqués sur une feuille annexe.

Cadre n° IV MANDATAIRE OU REPRÉSENTANT COMMUN; OU ADRESSE POUR LA CORRESPONDANCE

La personne dont l'identité est donnée ci-dessous est/a été désignée pour agir au nom du ou des déposants auprès des autorités internationales compétentes, comme: ☒ mandataire ☐ représentant commun

Nom et adresse : (Nom de famille suivi du prénom: pour une personne morale, désignation officielle complète. L'adresse doit comprendre le code postal et le nom du pays.)

BULL S.A  
Denis Hervé  
PC58D20/ 68, route de Versailles  
78434 LOUVECIENNES CEDEX

n° de téléphone  
(33) 1 39.66.61.80

n° de télécopieur  
(33) 1 39.66.61.73

n° de téléimprimeur

☐ Adresse pour la correspondance : cocher cette case lorsque aucun mandataire ni représentant commun n'est/n'a été désigné et que l'espace ci-dessus est utilisé pour indiquer une adresse spéciale à laquelle la correspondance doit être envoyée.

**THIS PAGE BLANK (USPTO)**

## Suite du cadre n° III AUTRE(S) DÉPOSANT(S) OU (AUTRE(S)) INVENTEUR(S)

Si aucun des sous-cadres suivants n'est utilisé, cette feuille ne doit pas être incluse dans la requête.

Nom et adresse : (Nom de famille suivi du prénom; pour une personne morale, désignation officielle complète. L'adresse doit comprendre le code postal et le nom du pays. Le pays de l'adresse indiquée dans ce cadre est l'Etat où le déposant a son domicile si aucun domicile n'est indiqué ci-dessous.)

**Sitbon Gérard**  
8 rue Roli  
75014 PARIS  
FRANCE

Cette personne est :

- ☐ déposant seulement  
☒ déposant et inventeur  
☐ inventeur seulement  
(Si cette case est cochée, ne pas remplir la suite.)

Nationalité (nom de l'Etat) :

FRANCE

Domicile (nom de l'Etat) :

FRANCE

Cette personne est déposant pour :

- ☐ tous les États désignés ☐ tous les États désignés sauf les États-Unis d'Amérique ☒ les États-Unis d'Amérique seulement ☐ les États indiqués dans le cadre supplémentaire

Nom et adresse : (Nom de famille suivi du prénom; pour une personne morale, désignation officielle complète. L'adresse doit comprendre le code postal et le nom du pays. Le pays de l'adresse indiquée dans ce cadre est l'Etat où le déposant a son domicile si aucun domicile n'est indiqué ci-dessous.)

**Ravon Jean-Michel**  
1 avenue du Colonel Fabien  
78210 ST-CYR L'ECOLE  
FRANCE

Cette personne est :

- ☐ déposant seulement  
☒ déposant et inventeur  
☐ inventeur seulement  
(Si cette case est cochée, ne pas remplir la suite.)

Nationalité (nom de l'Etat) :

FRANCE

Domicile (nom de l'Etat) :

FRANCE

Cette personne est déposant pour :

- ☐ tous les États désignés ☐ tous les États désignés sauf les États-Unis d'Amérique ☒ les États-Unis d'Amérique seulement ☐ les États indiqués dans le cadre supplémentaire

Nom et adresse : (Nom de famille suivi du prénom; pour une personne morale, désignation officielle complète. L'adresse doit comprendre le code postal et le nom du pays. Le pays de l'adresse indiquée dans ce cadre est l'Etat où le déposant a son domicile si aucun domicile n'est indiqué ci-dessous.)

Cette personne est :

- ☐ déposant seulement  
☐ déposant et inventeur  
☐ inventeur seulement  
(Si cette case est cochée, ne pas remplir la suite.)

Nationalité (nom de l'Etat) :

Domicile (nom de l'Etat) :

Cette personne est déposant pour :

- ☐ tous les États désignés ☐ tous les États désignés sauf les États-Unis d'Amérique ☐ les États-Unis d'Amérique seulement ☐ les États indiqués dans le cadre supplémentaire

Nom et adresse : (Nom de famille suivi du prénom; pour une personne morale, désignation officielle complète. L'adresse doit comprendre le code postal et le nom du pays. Le pays de l'adresse indiquée dans ce cadre est l'Etat où le déposant a son domicile si aucun domicile n'est indiqué ci-dessous.)

Cette personne est :

- ☐ déposant seulement  
☐ déposant et inventeur  
☐ inventeur seulement  
(Si cette case est cochée, ne pas remplir la suite.)

Nationalité (nom de l'Etat) :

Domicile (nom de l'Etat) :

Cette personne est déposant pour :

- ☐ tous les États désignés ☒ tous les États désignés sauf les États-Unis d'Amérique ☐ les États-Unis d'Amérique seulement ☐ les États indiqués dans le cadre supplémentaire

☐ D'autres déposants ou inventeurs sont indiqués sur une autre feuille annexe.

**THIS PAGE BLANK (USPTO)**



## Cadre n° V DÉSIGNATION D'ÉTATS

Les désignations suivantes sont faites conformément à la règle 4.9.a) (cocher les cases appropriées; une au moins doit l'être):

## Brevet régional

- ☐ AP Brevet ARIPO : GH Ghana, GM Gambie, KE Kenya, LS Lesotho, MW Malawi, SD Soudan, SZ Swaziland, UG Ouganda, ZW Zimbabwe et tout autre État qui est un État contractant du Protocole de Harare et du PCT
- ☐ EA Brevet eurasien : AM Arménie, AZ Azerbaïdjan, BY Bélarus, KG Kirghizistan, KZ Kazakhstan, MD République de Moldova, RU Fédération de Russie, TJ Tadjikistan, TM Turkménistan et tout autre État qui est un État contractant de la Convention sur le brevet eurasien et du PCT
- ☒ EP Brevet européen : AT Autriche, BE Belgique, CH et LI Suisse et Liechtenstein, CY Chypre, DE Allemagne, DK Danemark, ES Espagne, FI Finlande, FR France, GB Royaume-Uni, GR Grèce, IE Irlande, IT Italie, LU Luxembourg, MC Monaco, NL Pays-Bas, PT Portugal, SE Suède et tout autre État qui est un État contractant de la Convention sur le brevet européen et du PCT
- ☐ OA Brevet OAPI : BF Burkina Faso, BJ Bénin, CF République centrafricaine, CG Congo, CI Côte d'Ivoire, CM Cameroun, GA Gabon, GN Guinée, GW Guinée-Bissau, ML Mali, MR Mauritanie, NE Niger, SN Sénégal, TD Tchad, TG Togo et tout autre État qui est un État membre de l'OAPI et un État contractant du PCT (si une autre forme de protection ou de traitement est souhaitée, le préciser sur la ligne pointillée).

## Brevet national (si une autre forme de protection ou de traitement est souhaitée, le préciser sur la ligne pointillée):

- |  |   |
|--|---|
| <input type="checkbox"/> AL Albanie                                    | <input type="checkbox"/> LS Lesotho                               |
| <input type="checkbox"/> AM Arménie                                    | <input type="checkbox"/> LT Lituanie                              |
| <input type="checkbox"/> AT Autriche                                   | <input type="checkbox"/> LU Luxembourg                            |
| <input type="checkbox"/> AU Australie                                  | <input type="checkbox"/> LV Lettonie                              |
| <input type="checkbox"/> AZ Azerbaïdjan                                | <input type="checkbox"/> MD République de Moldova                 |
| <input type="checkbox"/> BA Bosnie-Herzégovine                         | <input type="checkbox"/> MG Madagascar                            |
| <input type="checkbox"/> BB Barbade                                    | <input type="checkbox"/> MK Ex-République yougoslave de Macédoine |
| <input type="checkbox"/> BG Bulgarie                                   |   |
| <input type="checkbox"/> BR Brésil                                     | <input type="checkbox"/> MN Mongolie                              |
| <input type="checkbox"/> BY Bélarus                                    | <input type="checkbox"/> MW Malawi                                |
| <input type="checkbox"/> CA Canada                                     | <input type="checkbox"/> MX Mexique                               |
| <input type="checkbox"/> CH et LI Suisse et Liechtenstein              | <input type="checkbox"/> NO Norvège                               |
| <input type="checkbox"/> CN Chine                                      | <input type="checkbox"/> NZ Nouvelle-Zélande                      |
| <input type="checkbox"/> CU Cuba                                       | <input type="checkbox"/> PL Pologne                               |
| <input type="checkbox"/> CZ République tchèque                         | <input type="checkbox"/> PT Portugal                              |
| <input type="checkbox"/> DE Allemagne                                  | <input type="checkbox"/> RO Roumanie                              |
| <input type="checkbox"/> DK Danemark                                   | <input type="checkbox"/> RU Fédération de Russie                  |
| <input type="checkbox"/> EE Estonie                                    | <input type="checkbox"/> SD Soudan                                |
| <input type="checkbox"/> ES Espagne                                    | <input type="checkbox"/> SE Suède                                 |
| <input type="checkbox"/> FI Finlande                                   | <input type="checkbox"/> SG Singapour                             |
| <input type="checkbox"/> GB Royaume-Uni                                | <input type="checkbox"/> SI Slovaquie                             |
| <input type="checkbox"/> GD Grenade                                    | <input type="checkbox"/> SK Slovaquie                             |
| <input type="checkbox"/> GE Géorgie                                    | <input type="checkbox"/> SL Sierra Leone                          |
| <input type="checkbox"/> GH Ghana                                      | <input type="checkbox"/> TJ Tadjikistan                           |
| <input type="checkbox"/> GM Gambie                                     | <input type="checkbox"/> TM Turkménistan                          |
| <input type="checkbox"/> HR Croatie                                    | <input type="checkbox"/> TR Turquie                               |
| <input type="checkbox"/> HU Hongrie                                    | <input type="checkbox"/> TT Trinité-et-Tobago                     |
| <input type="checkbox"/> ID Indonésie                                  | <input type="checkbox"/> UA Ukraine                               |
| <input type="checkbox"/> IL Israël                                     | <input type="checkbox"/> UG Ouganda                               |
| <input type="checkbox"/> IN Inde                                       | <input checked="" type="checkbox"/> US États-Unis d'Amérique      |
| <input type="checkbox"/> IS Islande                                    |   |
| <input type="checkbox"/> JP Japon                                      | <input type="checkbox"/> UZ Ouzbékistan                           |
| <input type="checkbox"/> KE Kenya                                      | <input type="checkbox"/> VN Viet Nam                              |
| <input type="checkbox"/> KG Kirghizistan                               | <input type="checkbox"/> YU Yougoslavie                           |
| <input type="checkbox"/> KP République populaire démocratique de Corée | <input type="checkbox"/> ZW Zimbabwe                              |
| <input type="checkbox"/> KR République de Corée                        |   |
| <input type="checkbox"/> KZ Kazakhstan                                 |   |
| <input type="checkbox"/> LC Sainte-Lucie                               |   |
| <input type="checkbox"/> LK Sri Lanka                                  |   |
| <input type="checkbox"/> LR Libéria                                    |   |

Cases réservées pour la désignation (aux fins d'un brevet national) d'États qui sont devenus parties au PCT après la publication de la présente feuille:

Déclaration concernant les désignations de précaution : outre les désignations faites ci-dessus, le déposant fait aussi conformément à la règle 4.9.b) toutes les désignations qui seraient autorisées en vertu du PCT, à l'exception de toute désignation indiquée dans le cadre supplémentaire comme étant exclue de la portée de cette déclaration. Le déposant déclare que ces désignations additionnelles sont faites sous réserve de confirmation et que toute désignation qui n'est pas confirmée avant l'expiration d'un délai de 15 mois à compter de la date de priorité doit être considérée comme retirée par le déposant à l'expiration de ce délai. (Pour confirmer une désignation, il faut déposer une déclaration contenant la désignation en question et payer les taxes de désignation et de confirmation. La confirmation doit parvenir à l'office récepteur dans le délai de 15 mois.)

**THIS PAGE BLANK (USPTO)**

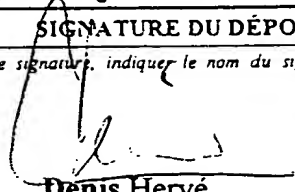
Cadre n° VI REVENDEICATION DE PRIORITÉ					
Date de dépôt de la demande antérieure (jour/mois/année)		Numéro de la demande antérieure	Lorsque la demande antérieure est une :		
			demande nationale : pays	demande régionale : * office régional	demande internationale : office récepteur
(1)	30 octobre 1998 (30.10.98)	98 13645	FRANCE		
(2)					
(3)					

☐ L'office récepteur est prié de préparer et de transmettre au Bureau international une copie certifiée conforme de la ou des demandes antérieures (seulement si la demande antérieure a été déposée auprès de l'office qui, aux fins de la présente demande internationale, est l'office récepteur) indiquées ci-dessus au(x) point(s) : 1

\* Si la demande antérieure est une demande ARIPO, il est obligatoire d'indiquer dans le cadre supplémentaire au moins un pays partie à la Convention de Paris pour la protection de la propriété industrielle pour lequel cette demande antérieure a été déposée (règle 4.10.b)ii)). Voir le cadre supplémentaire.

Cadre n° VII ADMINISTRATION CHARGÉE DE LA RECHERCHE INTERNATIONALE	
Choix de l'administration chargée de la recherche internationale (ISA) (si plusieurs administrations chargées de la recherche internationale sont compétentes pour procéder à la recherche internationale, indiquer l'administration choisie; le code à deux lettres peut être utilisé) : <b>ISA /</b>	Demande d'utilisation des résultats d'une recherche antérieure; mention de cette recherche (si une recherche antérieure a été effectuée par l'administration chargée de la recherche internationale ou demandée à cette dernière) : Date (jour/mois/année) : <b>30.10.98</b> Numéro : <b>98 13645</b> Pays (ou office régional) : <b>FR</b>

Cadre n° VIII BORDEREAU; LANGUE DE DÉPÔT	
La présente demande internationale contient le nombre de feuilles suivant :  requête : 04 description (sauf partie réservée au listage des séquences) : 45 revendications : 02 abrégé : 01 dessins : 06 partie de la description réservée au listage des séquences : _____  Nombre total de feuilles : 58	Le ou les éléments cochés ci-après sont joints à la présente demande internationale : 1. <input type="checkbox"/> feuille de calcul des taxes 2. <input checked="" type="checkbox"/> pouvoir distinct signé 1 3. <input type="checkbox"/> copie du pouvoir général; numéro de référence, le cas échéant : 4. <input type="checkbox"/> explication de l'absence d'une signature 5. <input type="checkbox"/> document(s) de priorité indiqué(s) dans le cadre n° VI au(x) point(s) : 6. <input type="checkbox"/> traduction de la demande internationale en (langue) : 7. <input type="checkbox"/> indications séparées concernant des micro-organismes ou autre matériel biologique déposés 8. <input type="checkbox"/> listage des séquences de nucléotides ou d'acides aminés sous forme déchiffrable par ordinateur 9. <input checked="" type="checkbox"/> autres éléments (préciser) : <b>Rapport de Recherche</b>
Figure des dessins qui doit accompagner l'abrégé : 01	Langue de dépôt de la demande internationale : <b>FRANCAIS</b>

Cadre n° IX SIGNATURE DU DÉPOSANT OU DU MANDATAIRE	
À côté de chaque signature, indiquer le nom du signataire et, si cela n'apparaît pas clairement à la lecture de la requête, à quel titre l'intéressé signe.  <div style="text-align: center;">   <b>Denis Hervé</b>  <b>(Mandataire)</b> </div>	

Réservé à l'office récepteur

1. Date effective de réception des pièces supposées constituer la demande internationale :	2. Dessins : <input type="checkbox"/> reçus :  <input type="checkbox"/> non reçus :
3. Date effective de réception, rectifiée en raison de la réception ultérieure, mais dans les délais, de documents ou de dessins complétant ce qui est supposé constituer la demande internationale :	
4. Date de réception, dans les délais, des corrections demandées selon l'article 11.2) du PCT :	
5. Administration chargée de la recherche internationale (si plusieurs sont compétentes) : <b>ISA /</b>	6. <input type="checkbox"/> Transmission de la copie de recherche différée jusqu'au paiement de la taxe de recherche.

Réservé au Bureau international

Date de réception de l'exemplaire original par le Bureau international :

**THIS PAGE BLANK (USPTO)**

PCT

AVIS INFORMANT LE DEPOSANT DE LA  
COMMUNICATION DE LA DEMANDE  
INTERNATIONALE AUX OFFICES DESIGNES

(règle 47.1.c), première phrase, du PCT)

Expéditeur: le BUREAU INTERNATIONAL

Destinataire:

DENIS, Hervé  
Bull S.A.  
68, route de Versailles  
PC58D20  
F-78434 Louveciennes Cedex  
FRANCE

Date d'expédition (jour/mois/année) 11 mai 2000 (11.05.00)		AVIS IMPORTANT	
Référence du dossier du déposant ou du mandataire PCT 3682/HD			
Demande internationale no PCT/FR99/02629	Date du dépôt international (jour/mois/année) 28 octobre 1999 (28.10.99)	Date de priorité (jour/mois/année) 30 octobre 1998 (30.10.98)	
Déposant BULL S.A. etc			

1. Il est notifié par la présente qu'à la date indiquée ci-dessus comme date d'expédition de cet avis, le Bureau international a communiqué, comme le prévoit l'article 20, la demande internationale aux offices désignés suivants:

US

Conformément à la règle 47.1.c), troisième phrase, ces offices acceptent le présent avis comme preuve déterminante du fait que la communication de la demande internationale a bien eu lieu à la date d'expédition indiquée plus haut, et le déposant n'est pas tenu de remettre de copie de la demande internationale à l'office ou aux offices désignés.

2. Les offices désignés suivants ont renoncé à l'exigence selon laquelle cette communication doit être effectuée à cette date:

EP

La communication sera effectuée seulement sur demande de ces offices. De plus, le déposant n'est pas tenu de remettre de copie de la demande internationale aux offices en question (règle 49.1)a-bis)).

3. Le présent avis est accompagné d'une copie de la demande internationale publiée par le Bureau international le 11 mai 2000 (11.05.00) sous le numéro WO 00/26774

**RAPPEL CONCERNANT LE CHAPITRE II (article 31.2)a) et règle 54.2)**

Si le déposant souhaite reporter l'ouverture de la phase nationale jusqu'à 30 mois (ou plus pour ce qui concerne certains offices) à compter de la date de priorité, la **demande d'examen préliminaire international** doit être présentée à l'administration compétente chargée de l'examen préliminaire international avant l'expiration d'un délai de 19 mois à compter de la date de priorité.

Il appartient exclusivement au déposant de veiller au respect du délai de 19 mois.

Il est à noter que seul un déposant qui est ressortissant d'un Etat contractant du PCT lié par le chapitre II ou qui y a son domicile peut présenter une demande d'examen préliminaire international.

**RAPPEL CONCERNANT L'OUVERTURE DE LA PHASE NATIONALE (article 22 ou 39.1))**

Si le déposant souhaite que la demande internationale procède en phase nationale, il doit, dans le délai de 20 mois ou de 30 mois, ou plus pour ce qui concerne certains offices, accomplir les actes mentionnés dans ces dispositions auprès de chaque office désigné ou élu.

Pour d'autres informations importantes concernant les délais et les actes à accomplir pour l'ouverture de la phase nationale, voir l'annexe du formulaire PCT/IB/301 (Notification de la réception de l'exemplaire original) et le volume II du Guide du déposant du PCT.

Bureau international de l'OMPI 34, chemin des Colombettes 1211 Genève 20, Suisse	Fonctionnaire autorisé  J. Zahra
no de télécopieur (41-22) 740.14.35	no de téléphone (41-22) 338.83.38

**THIS PAGE BLANK (USPTO)**

# PCT

## RAPPORT DE RECHERCHE INTERNATIONALE

(article 18 et règles 43 et 44 du PCT)

Référence du dossier du déposant ou du mandataire <b>PCT 3682/HD</b>	<b>POUR SUITE A DONNER</b> voir la notification de transmission du rapport de recherche internationale (formulaire PCT/ISA/220) et, le cas échéant, le point 5 ci-après	
Demande internationale n° <b>PCT/FR 99/ 02629</b>	Date du dépôt international (jour/mois/année) <b>28/10/1999</b>	(Date de priorité (la plus ancienne) (jour/mois/année) <b>30/10/1998</b>
Déposant <b>BULL S.A. et al.</b>		

Le présent rapport de recherche internationale, établi par l'administration chargée de la recherche internationale, est transmis au déposant conformément à l'article 18. Une copie en est transmise au Bureau international.

Ce rapport de recherche internationale comprend 3 feuilles.

☒ Il est aussi accompagné d'une copie de chaque document relatif à l'état de la technique qui y est cité.

**1. Base du rapport**

a. En ce qui concerne la langue, la recherche internationale a été effectuée sur la base de la demande internationale dans la langue dans laquelle elle a été déposée, sauf indication contraire donnée sous le même point.

☐ la recherche internationale a été effectuée sur la base d'une traduction de la demande internationale remise à l'administration.

b. En ce qui concerne les séquences de nucléotides ou d'acides aminés divulguées dans la demande internationale (le cas échéant), la recherche internationale a été effectuée sur la base du listage des séquences :

☐ contenu dans la demande internationale, sous forme écrite.

☐ déposée avec la demande internationale, sous forme déchiffrable par ordinateur.

☐ remise ultérieurement à l'administration, sous forme écrite.

☐ remise ultérieurement à l'administration, sous forme déchiffrable par ordinateur.

☐ La déclaration, selon laquelle le listage des séquences présenté par écrit et fourni ultérieurement ne vas pas au-delà de la divulgation faite dans la demande telle que déposée, a été fournie.

☐ La déclaration, selon laquelle les informations enregistrées sous forme déchiffrable par ordinateur sont identiques à celles du listage des séquences présenté par écrit, a été fournie.

2. ☐ Il a été estimé que certaines revendications ne pouvaient pas faire l'objet d'une recherche (voir le cadre I).

3. ☐ Il y a absence d'unité de l'invention (voir le cadre II).

**4. En ce qui concerne le titre,**

☒ le texte est approuvé tel qu'il a été remis par le déposant.

☐ Le texte a été établi par l'administration et a la teneur suivante:

**5. En ce qui concerne l'abrégé,**

☒ le texte est approuvé tel qu'il a été remis par le déposant

☐ le texte (reproduit dans le cadre III) a été établi par l'administration conformément à la règle 38.2b). Le déposant peut présenter des observations à l'administration dans un délai d'un mois à compter de la date d'expédition du présent rapport de recherche internationale.

**6. La figure des dessins à publier avec l'abrégé est la Figure n°**

☒ suggérée par le déposant.

☐ parce que le déposant n'a pas suggéré de figure.

☐ parce que cette figure caractérise mieux l'invention.

1  
☐ Aucune des figures n'est à publier.

**THIS PAGE BLANK (USPTO)**



**Demande Internationale No**  
**PCT/FR 99/02629**

page 1 de 2

**THIS PAGE BLANK (USPTO)**

C.(suite) DOCUMENTS CONSIDERES COMME PERTINENTS		
Catégorie	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
Y	SHOWMAN P S: "AN OBJECT-BASED USER INTERFACE FOR THE HP NEWWAVE ENVIRONMENT" HEWLETT-PACKARD JOURNAL, vol. 40, no. 4, 1 août 1989 (1989-08-01), pages 9-17, XP000094705 page 11, colonne de gauche, ligne 44 -page 12, colonne de droite, ligne 1	1
A	WO 96 37825 A (BULL HN INFORMATION SYST) 28 novembre 1996 (1996-11-28) abrégé page 3, ligne 15 -page 5, ligne 2; revendication 1	1,2

**THIS PAGE BLANK (USPTO)**

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/FR 99/02629

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
W0 9637825 A	28-11-1996	US 5678047 A EP 0827606 A	14-10-1997 11-03-1998

**THIS PAGE BLANK (USPTO)**

## NOTES RELATIVES AU FORMULAIRE PCT/ISA/220 (suite)

La lettre doit indiquer les différences existant entre les revendications telles que déposées et les revendications telles que modifiées. Elle doit indiquer en particulier, pour chaque revendication figurant dans la demande internationale (étant entendu que des indications identiques concernant plusieurs revendications peuvent être groupées), si

- i) la revendication n'est pas modifiée;
- ii) la revendication est supprimée;
- iii) la revendication est nouvelle;
- iv) la revendication remplace une ou plusieurs revendications telles que déposées;
- v) la revendication est le résultat de la division d'une revendication telle que déposée.

**Les exemples suivants illustrent la manière dont les modifications doivent être expliquées dans la lettre d'accompagnement:**

1. [Lorsque le nombre des revendications déposées initialement s'élevait à 48 et qu'à la suite d'une modification de certaines revendications il s'élève à 51]:  
"Revendications 1 à 15 remplacées par les revendications modifiées portant les mêmes numéros; revendications 30, 33 et 36 pas modifiées; nouvelles revendications 49 à 51 ajoutées."
2. [Lorsque le nombre des revendications déposées initialement s'élevait à 15 et qu'à la suite d'une modification de toutes les revendications il s'élève à 11]:  
"Revendications 1 à 15 remplacées par les revendications modifiées 1 à 11."
3. [Lorsque le nombre des revendications déposées initialement s'élevait à 14 et que les modifications consistent à supprimer certaines revendications et à en ajouter de nouvelles]:  
"Revendications 1 à 6 et 14 pas modifiées; revendications 7 à 13 supprimées; nouvelles revendications 15, 16 et 17 ajoutées." ou  
"Revendications 7 à 13 supprimées; nouvelles revendications 15, 16 et 17 ajoutées; toutes les autres revendications pas modifiées."
4. [Lorsque plusieurs sortes de modifications sont faites]:  
"Revendications 1-10 pas modifiées; revendications 11 à 13, 18 et 19 supprimées; revendications 14, 15 et 16 remplacées par la revendication modifiée 14; revendication 17 divisée en revendications modifiées 15, 16 et 17; nouvelles revendications 20 et 21 ajoutées."

### "Déclaration selon l'article 19.1)" (Règle 46.4)

Les modifications peuvent être accompagnées d'une déclaration expliquant les modifications et précisant l'incidence que ces dernières peuvent avoir sur la description et sur les dessins (qui ne peuvent pas être modifiés selon l'article 19.1)).

La déclaration sera publiée avec la demande internationale et les revendications modifiées.

**Elle doit être rédigée dans la langue dans laquelle la demande internationale est publiée.**

Elle doit être succincte (ne pas dépasser 500 mots si elle est établie ou traduite en anglais).

Elle ne doit pas être confondue avec la lettre expliquant les différences existant entre les revendications telles que déposées et les revendications telles que modifiées, et ne la remplace pas. Elle doit figurer sur une feuille distincte et doit être munie d'un titre permettant de l'identifier comme telle, constitué de préférence des mots "Déclaration selon l'article 19.1)".

Elle ne doit contenir aucun commentaire dénigrant relatif au rapport de recherche internationale ou à la pertinence des citations que ce dernier contient. Elle ne peut se référer à des citations se rapportant à une revendication donnée et contenues dans le rapport de recherche internationale qu'en relation avec une modification de cette revendication.

### Conséquence du fait qu'une demande d'examen préliminaire international ait déjà été présentée

Si, au moment du dépôt de modifications effectuées en vertu de l'article 19, une demande d'examen préliminaire international a déjà été présentée, le déposant doit de préférence, lors du dépôt des modifications auprès du Bureau international, déposer également une copie de ces modifications auprès de l'administration chargée de l'examen préliminaire international (voir la règle 62.2a), première phrase).

### Conséquence au regard de la traduction de la demande internationale lors de l'ouverture de la phase nationale

L'attention du déposant est appelée sur le fait qu'il peut avoir à remettre aux offices désignés ou élus, lors de l'ouverture de la phase nationale, une traduction des revendications telles que modifiées en vertu de l'article 19 au lieu de la traduction des revendications telles que déposées ou en plus de celle-ci.

Pour plus de précisions sur les exigences de chaque office désigné ou élu, voir le volume II du Guide du déposant du PCT.

**THIS PAGE BLANK (USPTO)**





## DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITE DE COOPERATION EN MATIÈRE DE BREVETS (PCT)

(51) Classification internationale des brevets <sup>7</sup> : <b>G06F 9/44</b>	<b>A1</b>	(11) Numéro de publication internationale: <b>WO 00/26774</b>
		(43) Date de publication internationale: 11 mai 2000 (11.05.00)

(21) Numéro de la demande internationale: PCT/FR99/02629

(22) Date de dépôt international: 28 octobre 1999 (28.10.99)

(30) Données relatives à la priorité:  
98/13645 30 octobre 1998 (30.10.98) FR(71) Déposant (pour tous les Etats désignés sauf US): BULL S.A.  
[FR/FR]; 68, route de Versailles, F-78434 Louveciennes  
Cedex (FR).

(72) Inventeurs; et

(75) Inventeurs/Déposants (US seulement): NACHEF, Armand  
[FR/FR]; 6, place Georges Pompidou, F-78180 Montigny  
le Bretonneux (FR). SITBON, Gérard [FR/FR]; 8, rue Roli,  
F-75014 Paris (FR). RAVON, Jean-Michel [FR/FR]; 1,  
avenue du Colonel Fabien, F-78210 Saint-Cyr l'Ecole (FR).(74) Mandataire: DENIS, Hervé; Bull S.A., 68, route de Versailles,  
PC58D20, F-78434 Louveciennes Cedex (FR).(81) Etats désignés: US, brevet européen (AT, BE, CH, CY, DE,  
DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

## Publiée

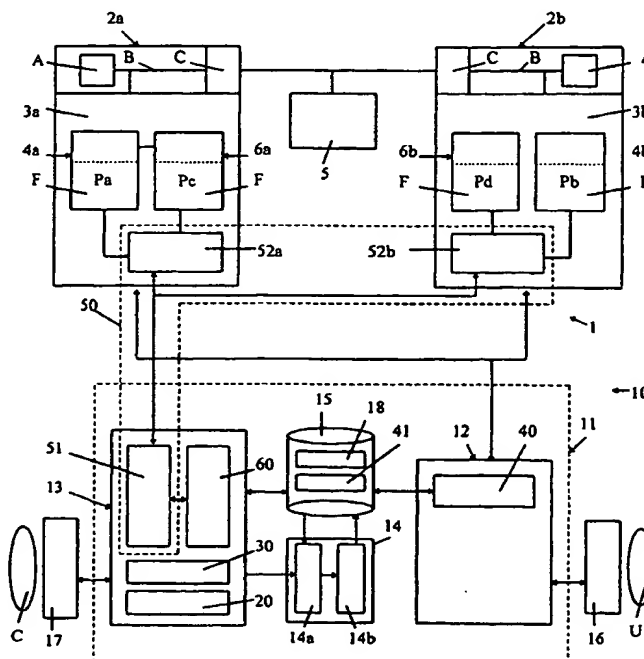
*Avec rapport de recherche internationale.**Avant l'expiration du délai prévu pour la modification des  
revendications, sera republiée si des modifications sont  
reçues.*(54) Title: METHOD FOR CONTROLLING A FUNCTION EXECUTABLE BY SPECIFIC COMMANDS TO DIFFERENT SOFTWARE  
TOOLS(54) Titre: PROCEDE DE COMMANDE D'UNE FONCTION EXECUTABLE PAR DES COMMANDES SPECIFIQUES A DES  
LOGICIELS DIFFERENTS

## (57) Abstract

The invention concerns a method for controlling a function (f) executable by various software tools (4a, 4b, 6a, 6b) by specific commands (Pa-Pd) to the respective software tools and each capable of having at least one option, the software tools being installed in at least a machine (2a, 2b) of a computer system (10). Said method consists in: defining in an abstract class an abstract method for the function (f); defining a common command (P0) including arbitrary symbols corresponding to parameters of the abstract method; forming at least a pilot (52) for implementing the abstract method in a machine; and causing the pilot to execute one of the specific commands with options equivalent to the options of the common command.

## (57) Abrégé

Le procédé sert à la commande d'une fonction (f) exécutable par divers logiciels (4a, 4b, 6a, 6b) sous des commandes (Pa-Pd) spécifiques aux logiciels respectifs et pouvant avoir chacune au moins une option, les logiciels étant installés dans au moins une machine (2a, 2b) d'un système informatique (10). Il consiste à définir dans une classe abstraite une méthode abstraite pour la fonction (f), à définir une commande commune (P0) incluant des symboles arbitraires correspondant aux paramètres de la méthode abstraite, à former au moins un pilote (52) pour implémenter la méthode abstraite dans une machine, et à faire exécuter par le pilote l'une des commandes spécifiques avec des options équivalant aux options de la commande commune.



### UNIQUEMENT A TITRE D'INFORMATION

Codes utilisés pour identifier les Etats parties au PCT, sur les pages de couverture des brochures publiant des demandes internationales en vertu du PCT.

AL	Albanie	ES	Espagne	LS	Lesotho	SI	Slovénie
AM	Arménie	FI	Finlande	LT	Lituanie	SK	Slovaquie
AT	Autriche	FR	France	LU	Luxembourg	SN	Sénégal
AU	Australie	GA	Gabon	LV	Lettonie	SZ	Swaziland
AZ	Azerbaïdjan	GB	Royaume-Uni	MC	Monaco	TD	Tchad
BA	Bosnie-Herzégovine	GE	Géorgie	MD	République de Moldova	TG	Togo
BB	Barbade	GH	Ghana	MG	Madagascar	TJ	Tadjikistan
BE	Belgique	GN	Guinée	MK	Ex-République yougoslave de Macédoine	TM	Turkménistan
BF	Burkina Faso	GR	Grèce	ML	Mali	TR	Turquie
BG	Bulgarie	HU	Hongrie	MN	Mongolie	TT	Trinité-et-Tobago
BJ	Bénin	IE	Irlande	MR	Mauritanie	UA	Ukraine
BR	Brésil	IL	Israël	MW	Malawi	UG	Ouganda
BY	Bélarus	IS	Islande	MX	Mexique	US	Etats-Unis d'Amérique
CA	Canada	IT	Italie	NE	Niger	UZ	Ouzbékistan
CF	République centrafricaine	JP	Japon	NL	Pays-Bas	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norvège	YU	Yougoslavie
CH	Suisse	KG	Kirghizistan	NZ	Nouvelle-Zélande	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	République populaire démocratique de Corée	PL	Pologne		
CM	Cameroun	KR	République de Corée	PT	Portugal		
CN	Chine	KZ	Kazakstan	RO	Roumanie		
CU	Cuba	LC	Sainte-Lucie	RU	Fédération de Russie		
CZ	République tchèque	LI	Liechtenstein	SD	Soudan		
DE	Allemagne	LK	Sri Lanka	SE	Suède		
DK	Danemark	LR	Libéria	SG	Singapour		
EE	Estonie						

**Titre.**

Procédé de commande d'une fonction exécutable par des commandes spécifiques à des logiciels différents.

**5 Domaine technique.**

L'invention a pour objet un procédé de commande d'une fonction exécutable par des commandes spécifiques à des logiciels différents dans un système informatique. Elle convient à tout système et est plus  
10 particulièrement adaptée à un système très hétérogène.

**L'art antérieur.**

Le système informatique d'une société est souvent hétérogène, en  
15 raison de l'évolution historique du système et des disponibilités du marché. Dans un système informatique se trouvent un ou plusieurs serveurs supportant plusieurs applications. L'hétérogénéité peut être due au fait que les serveurs ont des systèmes d'exploitation différents les uns des autres. Par exemple, l'un des serveurs peut être un système fermé, spécifique d'un  
20 constructeur donné et fonctionnant avec un système d'exploitation (Operating System) propre au constructeur, tandis que les autres serveurs peuvent être des systèmes ouverts fonctionnant avec des systèmes d'exploitation différents entre eux, tels que ceux connus sous les marques déposées UNIX et Windows NT. Chaque système d'exploitation possède des commandes spécifiques,  
25 écrites dans un langage spécifique. Par conséquent, l'exploitation d'un système informatique requiert l'emploi de commandes propres à chaque système d'exploitation et la connaissance de leurs langages.

L'hétérogénéité peut aussi provenir des applications. Plusieurs  
30 applications différentes entre elles peuvent avoir des parties fonctionnelles analogues exécutables au moyen d'interfaces de commande différentes entre elles. Par exemple, deux logiciels aussi différents qu'un logiciel de

comptabilité et un logiciel de gestion de base de données ont en commun la fonction d'impression de documents alors que leurs commandes sont tout à fait différentes.

5 Un utilisateur, que ce soit un administrateur ou un responsable d'exploitation du système informatique, doit donc connaître les différentes commandes des différents systèmes d'exploitation et des différents logiciels d'application. L'utilisateur doit donc avoir à la fois une solide formation relative à chaque système et chaque application et une bonne pratique pour  
10 les exploiter vite et efficacement. En outre, les systèmes d'exploitation et surtout les applications évoluent très vite et nécessitent une formation régulière des utilisateurs et un usage permanent des différents systèmes et applications. Les applications ayant des fonctions de plus en plus nombreuses et plus riches, le temps d'apprentissage est plus long. Les utilisateurs  
15 disposent souvent d'une documentation d'aide en ligne. Cependant, la documentation est aussi spécifique et évolutive, et elle est d'un emploi très lourd, surtout si les informations d'aide nécessitent l'ouverture et la lecture de plusieurs fenêtres successives.

## 20 **Sommaire de l'invention.**

Un but de l'invention est d'offrir dans un système informatique une commande commune à toutes les commandes représentatives d'une même fonction et spécifiques à divers logiciels, tels que des systèmes  
25 d'exploitation ou des applications.

L'invention a pour objet un procédé de commande d'une fonction exécutable par divers logiciels sous des commandes spécifiques aux logiciels respectifs et pouvant avoir chacune au moins une option, les logiciels étant  
30 installés dans au moins une machine d'un système informatique, caractérisé en ce qu'il consiste à définir dans une classe abstraite une méthode abstraite

-3-

pour la fonction, la méthode abstraite incluant des paramètres correspondant à l'union, au sens logique, de toutes les options des commandes spécifiques, à définir une commande commune incluant des symboles arbitraires correspondant aux paramètres de la méthode abstraite, à former au moins un pilote pour implémenter la méthode abstraite dans une machine, et à faire exécuter par le pilote l'une des commandes spécifiques avec des options équivalant aux options de la commande commune.

L'invention a pour objet corollaire un système informatique comprenant au moins un machine disposant de logiciels divers ayant en commun au moins une fonction exécutable sous des commandes spécifiques aux logiciels respectifs et pouvant avoir chacune au moins une option, caractérisé en ce qu'il met en œuvre le procédé défini précédemment.

L'invention a aussi pour objet corollaire un support d'enregistrement, tel qu'une disquette magnétique ou un cédérom, incorporant un logiciel pour la mise en œuvre du procédé.

Les caractéristiques et avantages de l'invention ressortent de la description qui suit, donnée à titre d'exemple et faite en référence aux dessins annexés.

## 20 Présentation des dessins.

♦ La figure 1 est une vue synoptique d'un système informatique mettant en œuvre le procédé de l'invention.

♦ La figure 2 est une fenêtre servant à un concepteur pour la mise en œuvre du procédé de l'invention.

♦ La figure 3 est une boîte de dialogue servant au concepteur pour créer dynamiquement une classe pour l'exploitation désirée du système par l'utilisateur.

♦ La figure 4 est une boîte de dialogue servant au concepteur pour définir un attribut de la classe présentée dans la figure 3.

-4-

♦ La figure 5 est une boîte de dialogue servant au concepteur pour définir une méthode de la classe présentée dans la figure 3.

♦ La figure 6 est une boîte de dialogue servant au concepteur pour définir un paramètre d'une méthode définie dans la figure 5.

5       ♦ La figure 7 est une vue synoptique de la structure en arbre de la classe telle que définie au moyen des boîtes de dialogue présentées dans les figures 3 à 6.

♦ La figure 8 est une vue synoptique illustrant un procédé de dérivation de la classe présentée dans la figure 7.

10       ♦ Les figures 9 et 10 sont des boîtes de dialogue servant au concepteur pour définir la classe dérivée en utilisant le procédé illustré dans la figure 8.

♦ Les figures 11 et 12 illustrent deux interfaces respectives servant à l'utilisateur pour la commande du système représenté sur la figure 1.  
15   1.

♦ La figure 13 est une fenêtre présentée à l'utilisateur pour l'utilisation des interfaces illustrées dans les figures 1, 11 et 12.

♦ La figure 14 est une boîte de dialogue servant au concepteur à la génération automatique dans un langage orienté objet d'un pilote  
20 d'implémentation d'une méthode abstraite d'une classe abstraite pour l'exécution d'une commande spécifique dans le système de la figure 1.

### **Description détaillée d'exemples illustrant l'invention.**

25       La figure 1 illustre un système d'information 10 exploité par un utilisateur U par l'intermédiaire d'une interface de commande 11. Le système 10 peut être un système quelconque. Le système 10 illustré inclut un ensemble informatique 1 comprenant au moins une machine, deux machines 2a et 2b dans l'exemple illustré. Chaque machine illustrée a, de façon bien  
30 connue, au moins un processeur A connecté par l'intermédiaire d'un bus B à une mémoire 3. D'une manière générale, la mémoire peut être locale ou distribuée, ou les machines peuvent former des noeuds d'un réseau dans

l'ensemble 1. Dans chaque mémoire 3 sont stockés des moyens logiciels, notamment un ou plusieurs systèmes d'exploitation 4. Afin de mieux mettre en relief les avantages de l'invention, il sera considéré comme un système hétérogène représentatif d'un cas difficile à gérer. Les deux machines 2 sont  
5 supposées fonctionner sous deux systèmes d'exploitation respectifs 4a et 4b de types ou de versions différents l'un de l'autre, par exemple un système d'exploitation du type propriétaire et un système d'exploitation de type ouvert, par exemple celui connu sous l'un des noms de marque déposée UNIX ou Windows NT. L'ensemble informatique 1 dispose aussi de moyens de base  
10 de données, appelés base de données 5 et pouvant être constitués d'une ou plusieurs bases de données locales ou distantes. Les machines 2 ont aussi chacune une interface d'entrée-sortie C connectée au bus B. Les interfaces d'entrée-sortie C des deux machines sont interconnectées entre elles ainsi qu'à la base de données C.

15

L'interface de commande 11 comprend un module de commande 12, un module de conception 13, un générateur 14 d'interfaces et une base d'interface 15. Le module 12 est connecté de façon bidirectionnelle à l'ensemble informatique 1. Le module 13 peut aussi être connecté de la même  
20 façon à l'ensemble informatique 1. La base d'interface 15 est connectée de façon bidirectionnelle aux modules 12 et 13 et au générateur 14. Le générateur 14 est aussi connecté pour être commandé par le module 13. Le module de commande 12 sert à l'utilisateur U pour commander et exploiter les données du système 10. Le module de conception 13 sert à un concepteur  
25 C, qui est un autre utilisateur de l'interface 11 mais qui est un spécialiste en informatique. Les deux utilisateurs U et C ont des consoles respectives 16 et 17 attachées aux modules respectifs 12 et 13.

En bref, l'utilisateur U définit des besoins pour connaître et  
30 améliorer le fonctionnement de l'entreprise et soumet ses besoins au concepteur C. Le concepteur C crée au moyen de son module 13 des objets logiciels répondant aux besoins de l'utilisateur U et les envoie au générateur

14 pour être converties en interfaces 18 (figures 11 et 12) correspondant à ses besoins et exprimant les commandes du système en un langage qui lui est compréhensible. Ces interfaces servent à l'utilisateur pour former des documents et seront appelées interfaces documentaires.

5

L'utilisateur U sera par exemple un directeur d'agence bancaire non spécialisé en informatique. On suppose par exemple que l'utilisateur U désire consulter l'ensemble informatique 1 de la banque pour faire des requêtes en vue d'obtenir une liste d'étudiants, clients de l'agence, et  
10 imprimer pour eux des lettres personnalisées d'information, d'invitation et de rappel. L'utilisateur U transmet ses besoins au concepteur C, par exemple lors d'une réunion entre eux. Le concepteur transcrit ces besoins en utilisant, dans une première étape, le module 13 pour créer, modifier et/ou supprimer des objets et des classes d'objets concernés par ces besoins. Par exemple, afin  
15 de satisfaire au besoin précité de l'utilisateur U, le concepteur C créera une classe impression avec : un premier attribut "liste de diffusion" selon un ensemble de critères, ici les étudiants de l'agence ; un second attribut "nombre d'exemplaires" (entier) ; une première méthode "imprimer" ayant comme paramètre l'imprimante destinataire ; et une seconde méthode  
20 "visualiser les opérations d'impression" pour connaître le déroulement de l'impression. Les objets et classes créés par le concepteur sont stockés dans la base d'interface 15. La transcription finale par le concepteur C des besoins de l'utilisateur U n'est contenue que dans des objets qui ont été directement créés précédemment ou qui sont des instances de classes définies  
25 précédemment. Ces objets sont contenus dans la base 15, qui est contenue dans l'interface de commande 11 dans l'exemple illustré mais qui pourrait être extérieure à l'interface 11 et incluse dans l'ensemble informatique 1. Les objets sont convertis en interfaces documentaires présentées à l'utilisateur U pour lui servir à former des documents de commande satisfaisant à ses  
30 besoins.



Le module 13 va d'abord être présenté. Il met en œuvre deux outils logiciels 20 et 30 exécutant respectivement un procédé de création dynamique de classes et un procédé de dérivation de classes et objets. L'outil 20 de création dynamique de classes va maintenant être présenté.

5

La figure 2 illustre un exemple d'écran 21 qu'affiche la console 17 pour servir d'interface entre le concepteur C et le module 13. L'écran 21 a une barre 22 de menus incluant notamment les menus "Classes", "Objets", "Rechercher" et "Exporter à U". Le menu "Classes" permet de créer dynamiquement des classes d'objets, par héritage ou non. Il permet aussi d'ouvrir une classe pour éventuellement la modifier, l'enregistrer ou la fermer. Le menu "Objets" permet d'instancier les classes définies précédemment, afin de créer des objets qu'on peut ensuite cloner. Comme pour les classes, on peut ouvrir un objet, le modifier, l'enregistrer sur le disque et le fermer. Avec l'outil 20, le concepteur C peut définir les commandes qui se trouvent dans les deux menus tels qu'illustrés par exemple dans ceux de l'écran 21. Le menu "Classes" illustré contient les commandes "Nouvelle classe", "Nouvelle classe héritée", "Ouvrir une classe", "Supprimer une classe", "Enregistrer la classe", "Enregistrer la classe sous..." (Save the class as...) et "Fermer la classe". Le menu "Objets" contient les commandes "Nouvel objet de classe", "Cloner un objet", "Ouvrir un objet", "Supprimer un objet", "Enregistrer l'objet", "Enregistrer l'objet sous..." et "Fermer l'objet".

L'outil 20 met en œuvre un procédé de génération dynamique d'une classe d'objets. Le procédé est déclenché automatiquement par le module de conception 13 lorsque le concepteur C active la commande "Nouvelle classe" du menu "Classes" de la fenêtre 22 de la figure 2. Le procédé de génération automatique d'une classe consiste à créer une classe générique globale "GenericClass" ayant deux membres possibles, l'un étant relatif à au moins un attribut et l'autre étant relatif à au moins une méthode, au moins l'un des deux membres étant une instance d'une classe générique, et à instancier la classe générique globale pour avoir ladite classe d'objets. Dans

l'outil 20 illustré, deux classes génériques "GenericAttribute" et "GenericMethod" sont attribuées aux deux membres respectifs de la classe générique globale "GenericClass". En outre, si une méthode inclut au moins un paramètre non fixe, ce paramètre est lui même une instance d'une classe générique correspondante "GenericParameter".

D'une manière générale, les quatre classes génériques ainsi créées ne sont pas visibles par le concepteur C. Dans l'exemple illustré, elles sont mises à la disposition du concepteur C lorsqu'il exécute une commande dans les menus "Classes" et "Objets". Ainsi, une classe générique se définit comme une classe permettant au concepteur C de créer une nouvelle classe en créant une instance d'une classe générique globale. Puisque la création d'instances est dynamique dans tous les langages, la création d'une classe devient aussi dynamique pour le concepteur C. De même, sachant qu'un objet est une nouvelle instance d'une classe, la création d'un objet correspond alors aussi à la création d'une copie de cette classe, c'est-à-dire à une nouvelle instance d'une classe générique globale. Par conséquent, le procédé déclenché par l'outil 20 est aussi déclenché lorsque le concepteur C active la commande "Nouvel objet de classe" dans le menu "Objets" de la fenêtre 22 de la figure 2. Le procédé va maintenant être illustré de la façon qu'il se présente au concepteur C par l'intermédiaire de l'outil 20.

La figure 3 illustre un exemple de boîte de dialogue 23 que l'outil 20 ouvre lorsque le concepteur C clique sur la commande "Nouvelle classe" du menu "Classes". Le concepteur C utilise la boîte de dialogue 23 pour saisir toutes les données relatives à une nouvelle classe qui n'hérite pas. On sait que les données sont les attributs et les méthodes de la nouvelle classe.

La boîte 23 illustrée contient, pour la saisie des données,

- un champ de texte "Nom de la classe"
- un champ de texte "Identification de la classe (Class Id)"

- une zone de texte "Description de la classe"
- une liste "Méthodes de la classe" et
- une liste "Attributs de la classe"

5                    La boîte 23 contient aussi les six boutons de commande encadrés en trait épais :

- "Modifier la méthode"
- "Supprimer la méthode"
- "Ajouter une méthode"
- 10                   - "Modifier l'attribut"
- "Supprimer l'attribut" et
- "Ajouter un attribut".

                  Lorsque cette boîte de dialogue est validée, elle se traduit par la  
15                   création d'une instance d'une classe générique globale, dite "GenericClass".  
La classe générique globale de l'exemple illustré dans la figure 3 a plusieurs attributs.

                  Le premier attribut est un nom, formé par une chaîne de  
20                   caractères désignant le nom de la classe. Il s'écrit en remplissant le champ  
"Nom de la classe" de la boîte 23.

                  Le second attribut est une identification de la classe, formée par  
une chaîne de caractères permettant d'identifier de façon unique la classe ou  
25                   l'objet dans le système. Cet attribut s'écrit en remplissant le champ  
"Identification de la classe" dans la boîte 23, par exemple en indiquant la  
date, l'heure de création, un nombre aléatoire formant un ticket, etc.

                  Le troisième attribut est une description, formée par du texte qui  
30                   décrit la raison d'être de la classe. Il s'écrit en remplissant la zone  
"Description de la classe".

Le quatrième attribut est un tableau nommé "Attributs", qui référence les attributs de la classe. Les attributs de la classe sont eux-mêmes des instances d'une classe générique d'attributs, dite "GenericAttribute", qui a comme attributs :

- le nom de l'attribut,
- la description de l'attribut,
- soit le type de l'attribut, soit les valeurs possibles de l'attribut,
- la valeur par défaut de l'attribut, donnée à la création de la classe "GenericAttribute", et
- la valeur réelle de l'attribut, qui est invisible au concepteur C et qui n'est donc pas définie à la création de la classe. Elle est définie par l'utilisateur U de la façon décrite ultérieurement.

Par exemple, la classe "Attributs[i]" est une instance de "GenericAttribute" et référence le ième attribut de la classe définie par le concepteur C.

La figure 4 illustre une boîte de dialogue 24 de l'outil 20 constitutif du quatrième attribut de "GenericClass". Cette boîte s'ouvre à partir de la boîte 23, par exemple en nommant "Classe1" la classe issue par instanciation de la classe générique globale "GenericClass" et en activant le bouton "Ajouter un attribut". La boîte 24 contient :

- un champ de texte "Nom de l'attribut",
- une zone de texte "Description de l'attribut",
- un champ de texte "Type de l'attribut",
- une zone de texte pour "Valeurs possibles de l'attribut",
- un champ de texte "Valeur par défaut de l'attribut", et
- un champ de texte "Valeur réelle de l'attribut" représenté en

grisé sur la figure 4 pour indiquer qu'il n'est pas visible par le concepteur C.

-11-

De même, pour modifier un attribut, il suffit de sélectionner l'attribut dans la liste des attributs de la figure 3 et d'activer le bouton "Modifier l'attribut", de façon à avoir la boîte de dialogue 24 de la figure 4 et à modifier les données qui apparaissent dans la boîte 24.

5

En tant que premier attribut, on donne à l'attribut le nom "Attribut1" et on remplit les autres champs. La validation de la boîte de dialogue 24 se traduit par la création d'une instance de la classe générique "GenericAttribute". Le nom "Attribut1" va figurer dans la liste des attributs de la figure 3, comme indiqué.

10

Le cinquième attribut de la classe générique "GenericClass" est un tableau nommé "Méthodes", qui référence les méthodes de la classe. Ces méthodes sont aussi, dans l'exemple illustré, des instances d'une classe générique de méthodes dite "GenericMethod". Une méthode est définie par les attributs suivants :

15

- son nom,
- sa description,
- son type de retour,
- ses paramètres, et
- son script d'exécution.

20

La figure 5 illustre une boîte de dialogue 25 de l'outil 20 constitutif du cinquième attribut de "GenericClass". Cette boîte s'ouvre à partir de la boîte 23, par exemple en activant le bouton "Ajouter une méthode". La boîte 25 contient :

25

- un champ de texte "Nom de la méthode",
- une zone de texte "Description de la méthode",
- un champ de texte "Type de retour de la méthode",
- une liste de "Paramètres de la méthode",
- un bouton de commande "Ajouter",

30

-12-

- un bouton de commande "Modifier", et
- un bouton de commande "Supprimer".

En tant que première méthode, on donne un nom à la méthode, par exemple "Méthode1", et on remplit les autres champs. La validation de la boîte de dialogue 25 se traduit par la création d'une instance de la classe générique "GenericMethod". Le nom "Méthode1" va figurer dans la liste des méthodes de la figure 3, comme indiqué.

La classe générique "GenericMethod" contient, en plus des attributs de la méthode, un tableau "Paramètres", qui référence les paramètres de la méthode. Le tableau peut être vide si la classe générique ne contient pas de paramètre à déterminer, ou peut ne pas exister si cette classe générique n'est destinée à n'avoir aucun paramètre déterminable ou si elle n'a que des paramètres fixes ou prédéterminés. Dans le tableau, les paramètres sont eux aussi des instances d'une classe générique "GenericParameter". Un paramètre d'une méthode est défini par ses attributs suivants :

- son nom,
- sa description,
- soit son type, soit ses valeurs possibles,
- sa valeur par défaut, et
- sa valeur réelle, non visible du concepteur C.

Par exemple, si le paramètre se rapporte au nombre de pages à imprimer, la valeur par défaut sera 1 mais l'utilisateur U pourra indiquer une valeur réelle autre, par exemple 3 pour avoir trois exemplaires des pages à imprimer.

La figure 6 illustre une boîte de dialogue 26 de l'outil 20 pour remplir la table de paramètres de "GenericMethod" dans la figure 5. La boîte 26 s'ouvre à partir de la boîte 25, en activant le bouton "Ajouter". La boîte 26 contient :

-13-

- un champ de texte "Nom du paramètre",
- une zone de texte "Description du paramètre",
- un champ de texte "Type du paramètre",
- une zone de texte "Valeurs possibles du paramètre",
- 5 - un champ de texte "Valeur par défaut du paramètre", et
- un champ de texte "Valeur réelle du paramètre" représenté en grisé sur la figure 5 pour indiquer que ce champ n'est pas visible du concepteur C.

10           En tant que premier paramètre, on lui donne par exemple le nom "Param1" et on remplit les autres champs. La validation de la boîte de dialogue 26 se traduit par la création d'une instance de la classe générique "GenericParameter". Le nom "Param1" va figurer dans la liste des paramètres de la figure 5.

15           La boîte de dialogue 23 de la figure 3 étant remplie, la classe "Classe1" est créée dynamiquement. On peut la valider en la sauvegardant avec la commande "Enregistrer la classe" ou "Enregistrer la classe sous...".

20           La figure 7 illustre la structure en arbre 27a de la classe "Classe1" résultant de la validation de la boîte 23. Dans la figure 7, la classe "Classe1" est une instance de la classe générique globale "GenericClass" et a un attribut "Attribut1" et une méthode "Méthode1" ayant elle-même un paramètre "Param1", qui sont tous les trois des instances de trois classes  
25 génériques respectives. Bien sûr, une autre classe, instance de la classe générique globale "GenericClass", pourrait avoir plusieurs instances de chacune des trois classes génériques "GenericAttribute", "GenericMethod" et "GenericParameter". Cette structure en arbre permet de dynamiquement modifier, ajouter et supprimer à tout moment des membres de la classe  
30 (attributs ou méthodes).

On sait qu'un objet est une instance d'une classe. L'activation de la commande "Nouvel objet de classe" dans le menu "Objets" de la fenêtre 21 de la figure 2 affiche sur la console 17 du concepteur C une boîte de dialogue (non illustrée) qui contient la liste de toutes les classes déjà définies. Le concepteur C peut en sélectionner une, qui sera la classe du nouvel objet. Dans un objet on donne éventuellement des valeurs aux attributs. Ces valeurs représenteront son identité et son état.

En outre, dans la classe générique globale "GenericClass" telle qu'elle apparaît de la figure 3 a de préférence un attribut supplémentaire consistant en un booléen "0" ou "1" indiquant si l'instance de la classe générique en cours représente une classe ou un objet. Dans l'exemple illustré, le booléen est "1" pour indiquer que le concepteur crée une classe. Le booléen se met automatiquement à la valeur correspondante en réponse à la commande "Nouvelle classe" ou "Nouvel objet de classe" qui a été actionnée par le concepteur C dans la fenêtre 22 de la figure 2. Dans l'exemple choisi, cet attribut n'est pas visible du concepteur C et est donc représenté en grisé.

D'autre part, dans la fenêtre 21 de la figure 2, le menu "Rechercher" est très utile au concepteur C pour faire une recherche puissante dans la base 15 pour y trouver des classes et des objets créés selon une liste de critères qu'il définit. Le concepteur C peut, par exemple, créer une nouvelle classe par héritage à partir de la classe qu'il a trouvée, au lieu de partir de zéro et refaire un travail déjà accompli.

25

Plus généralement, les boîtes de dialogue 23-26 servent donc au concepteur pour générer respectivement la classe générique globale et les classes génériques qui peuvent la composer. Bien que les boîtes de dialogue illustrées définissent complètement ces classes, tous les attributs, les types, les champs et zones de texte illustrés ne sont pas nécessaires. En particulier, les descriptions sont illustrées à une fin autre que le procédé de génération automatique de classes qui vient d'être décrit.



La description va maintenant porter sur l'outil 30 de dérivation d'une classe ou d'un objet, l'outil 30 étant contenu dans le module 13 de l'interface de commande 11 de la figure 1.

L'héritage avec la commande "Nouvelle classe héritée", l'instanciation avec la commande "Nouvel objet de classe" et le clonage avec la commande "Cloner un objet" nécessitent en commun la création dynamique d'une copie de la classe (pour l'héritage ou l'instanciation) ou d'une copie de l'objet (pour le clonage).

La figure 7 illustre aussi un arbre 27b d'une classe "ClasseD1" obtenue par duplication de "Classe1". Cette figure illustre, d'une manière générale, que la duplication de l'instance de "GenericClass" ne suffit pas, car la classe dupliquée "ClasseD1" va pointer aux mêmes instances "GenericMethod" et "GenericAttribute" de l'instance "Classe1" de "GenericClass". Par exemple, il ressort de la figure 7 que "Attribut1" est commun aux deux classes, alors que les deux classes devraient être distinctes et indépendantes. Par conséquent, pour dériver une classe ou un objet, c'est-à-dire pour créer par héritage une nouvelle classe, pour instancier une classe ou pour cloner un objet, il faut non seulement dupliquer l'instance de "GenericClass", mais aussi dupliquer chaque instance référencée directement ou indirectement par l'instance de "GenericClass".

25

D'autre part, les développeurs de logiciels utilisent une technique de sérialisation qui s'applique à de nombreux langages, tels que Java ou C++. La sérialisation permet de stocker tout objet sur un support d'enregistrement de données, un disque par exemple. Si par exemple un premier objet référence des objets, qui eux-mêmes référencent d'autres objets et ainsi de suite, il suffit d'écrire le premier objet sur le disque avec la méthode de sérialisation pour stocker automatiquement en mémoire

30

l'arborescence de tous les objets référencés directement ou indirectement par le premier objet.

La figure 8 illustre schématiquement le principe du procédé de  
5 dérivation d'une classe "Classe1" ou d'un objet. L'outil 30 met en œuvre ce  
procédé. L'exemple illustré porte sur la création d'une classe "ClasseD2" par  
héritage de la classe "Classe1" telle qu'illustrée dans la figure 7, étant bien  
entendu que le procédé peut s'appliquer de la même façon à l'instanciation  
d'une classe ou le clonage d'un objet. Comme indiqué à la figure 8, le procédé  
10 de dérivation consiste en une première étape à sérialiser la classe "Classe1"  
en la copiant sur un disque D, par exemple le disque dur d'une machine 2a ou  
2b de l'ensemble informatique 1 de la figure 1. Ainsi, toute l'arborescence 27a  
de cette classe, telle qu'illustrée dans la figure 7, va être dupliquée et stockée  
dans le disque. La seconde étape consiste à lire la classe dupliquée, stockée  
15 dans le disque D, en la chargeant dans l'outil 30, c'est-à-dire dans la mémoire  
vive sur laquelle est chargé le programme de l'outil. Il en résulte la classe  
dupliquée "ClasseD2" ayant une arborescence 27c identique à celle de  
l'arborescence 27a de la classe mère "Classe1" mais indépendante de la classe  
mère. En langage Java par exemple, les deux étapes du procédé peuvent  
20 s'exécuter par exemple par les deux instructions suivantes :

```
FileStream.writeObject(Classe1) ;  
// pour sérialiser l'arborescence d'origine Classe1  
Nouvel objet ou Nouvelle classe = FileStream.readObject() ;  
25 // pour avoir une copie de Classe1
```

La figure 9 illustre une boîte de dialogue 31 pour l'utilisation de  
l'outil 30. Cet exemple se rapporte à celui de la figure 9, qui porte sur la  
création d'une classe héritant d'une autre classe. En exécutant la commande  
30 "Nouvelle classe héritée" du menu "Classes" de la figure 3, la fenêtre de  
dialogue 31 de la figure 9 apparaît. Cette fenêtre contient une zone de texte  
"Nom de la nouvelle classe" et une liste des classes parmi lesquelles la classe

mère peut être choisie. Dans l'exemple choisi, la classe "Classe1" de la liste est choisie, comme indiqué en grisé à la figure 9. Dans l'exemple précédent, la classe héritée se nomme "classeD2".

5                   La figure 10 illustre un exemple de boîte de dialogue 32 résultant de la validation de la boîte 31. La boîte 32 est semblable à la boîte 23 de la figure 3. Par conséquent, seules les différences vont être indiquées ci-après. Les données contenues dans la boîte 32 sont remplies automatiquement par le logiciel de l'outil 30. Le nom de la classe est celui qui  
10 a été indiqué par le concepteur C dans la boîte 31 de la figure 9. La boîte 32 contient en plus un tableau indiquant le ou les parents de la classe, ici "Classe1" que le concepteur C a sélectionnée dans la liste de la boîte 31. Les autres données sont semblables à celles de la boîte 23, puisque l'arborescence 27c de la classe fille "ClasseD2" est la même que l'arborescence 27a de la  
15 classe mère "Classe1". Grâce aux boutons de commande de la boîte 32, il est possible de faire évoluer simplement la nouvelle classe "ClasseD2" indépendamment de la classe mère "Classe1".

La classe générique globale "GenericClass" a donc un attribut  
20 supplémentaire nommé "parent", qui est un tableau contenant des instances de la classe générique globale, telles que Classe1. Le tableau permet de reconnaître la provenance de l'instance en cours de la façon suivante :

- (a) Si l'instance en cours est un objet, alors :
  - si d'une manière générale "parent[0]" (indiqué par  
25 exemple dans le tableau auquel se réfère l'instance en cours) est un objet, l'objet en cours a été cloné à partir de parent[0],
  - si "parent[0]" est une classe, l'objet en cours est une instance de parent[0],
- 30 (b) Si l'instance en cours est une classe, alors "parent[0]" est

-18-

- soit vide, ce qui veut dire que la classe a été créée sans héritage avec la commande "Nouvelle classe",
- soit une classe (et non un objet), ce qui veut dire que la classe en cours a hérité de "parent[0]" grâce à la commande "Nouvelle classe héritée"

(c) Il en résulte, par itération, que le tableau "parent" indique tous les ascendants de l'instance en cours.

Le procédé ressort clairement des figures 8, 9 et 10. En réponse à la première instruction indiquée dans la figure 8, "Classe1" est sérialisée en la copiant sur le disque D. En réponse à la seconde instruction, la classe ainsi sérialisée est sauvegardée de nouveau en mémoire, mais avec modifications d'attributs, notamment le nom de classe "ClasseD2" et le tableau "Parent". Les deux arborescences 27a et 27c sont donc les mêmes, mais elles sont séparées l'une de l'autre. Par exemple, bien qu'elles aient les deux mêmes instances "Attribut1", en réalité ces deux instances sont complètement distinctes l'une de l'autre en mémoire et peuvent vivre indépendamment l'une de l'autre. Notamment, elles peuvent à tout moment être modifiées différemment l'une de l'autre. Cet exemple fait aussi ressortir qu'il en est de même pour les deux autres commandes "Nouvel objet de classe" et "Cloner un objet".

Plus généralement, il ressort de la description qui précède que le procédé de dérivation d'une classe d'objets et/ou d'un objet ayant un nom donné consiste à faire une copie de tout l'arbre de la classe ou de l'objet, à mémoriser la copie de l'arbre et à changer ledit nom pour attribuer un nom à la copie mémorisée. On a vu que de préférence la copie est faite par une sérialisation de l'arbre représentatif de ladite classe ou dudit objet en copiant l'arbre dans une mémoire D et la mémorisation de la copie de l'arbre consiste à la copier à nouveau en mémoire 30. En outre, on a vu aussi que la sérialisation, qui peut se faire dans divers langages, est particulièrement simple à mettre en œuvre en langage Java®.

La fonction du générateur 14 d'interface va maintenant être décrite. Jusqu'à présent, on a vu comment le concepteur C peut facilement  
5 créer des classes d'objets et des objets répondant aux besoins de l'utilisateur U. Les classes et objets ainsi créés sont stockés dans la base d'interface 15. Cependant, ces classes et objets demeurent incompréhensibles et inexploitable pour l'utilisateur U. De préférence, il est d'ailleurs interdit à l'utilisateur U d'y avoir accès, afin d'assurer au concepteur leur intégrité. Le  
10 générateur 14 a pour fonction de transformer les classes et objets ainsi créés en des interfaces sous forme de documents dans lesquels les commandes répondant aux besoins exprimés par l'utilisateur U sont compréhensibles par lui. Afin de distinguer ces interfaces des autres interfaces impliquées dans cette description, elles seront appelées interfaces documentaires. Le langage  
15 employé dans ces interfaces documentaires peut être le langage courant et/ou un langage spécialisé de la compétence de l'utilisateur.

Le fonctionnement du générateur 14 va ressortir de l'exemple qui suit, qui reprend et explicite l'exemple précédent selon lequel l'entreprise est  
20 une agence bancaire, l'utilisateur U est le directeur de l'agence et le concepteur C est un informaticien de la banque. On suppose que le directeur U veut : (1) consulter le système d'information de la banque pour faire des requêtes dans les bases de données en vue d'obtenir une liste de ses clients d'un type donné, des étudiants dans l'exemple considéré, et (2) imprimer des  
25 lettres personnalisées de plusieurs types, telles que des lettres d'information, des lettres d'invitation et des lettres de rappel. L'informaticien C traduit l'exploitation désirée par le directeur en deux questions. En l'occurrence, l'informaticien crée en réponse deux classes d'objets : "consultation\_système" et "impression" au moyen des deux outils 20 et 30. Il peut créer chacune de  
30 ces deux classes directement en activant la commande "Nouvelle classe" de la fenêtre 22 de la figure 2, soit indirectement par dérivation d'une classe semblable. Dans ce dernier cas, le concepteur C pourra actionner la

commande "Rechercher" de la fenêtre 22 pour trouver une classe semblable. Si une classe semblable existe, il la sélectionne dans la liste des classes et peut la dériver, par exemple en activant la commande "Nouvelle classe héritée" de la figure 2 et en modifiant, ajoutant ou supprimant des attributs et/ou des méthodes.

Le générateur 14 met en œuvre un procédé de génération d'une interface documentaire pour la commande d'un système informatique par un utilisateur, la commande étant faite à partir d'au moins un objet incluant des descriptions. Le procédé consiste à extraire de l'objet au moins en partie lesdites descriptions et à les organiser pour traduire le sens de ladite commande en un langage compréhensible à l'utilisateur et ainsi faire de ladite interface une interface documentaire. Le générateur 14 comprend donc un bloc d'extraction 14a et un bloc de génération 14b. Le bloc d'extraction 14a prélève de la base d'interface 15 l'objet sélectionné par le concepteur et y extrait les descriptions. Le bloc de génération 14b organise les descriptions pour former l'interface documentaire et la stocker dans la base d'interface 15.

Le procédé va être mieux compris à partir de l'exemple précédent. Parmi les attributs de la classe "consultation\_système" de l'exemple considéré se trouvent des méthodes qui font appel à des commandes spécifiques pour l'exploitation des données du système, en l'occurrence aux bases de données du système 10. Parmi ces méthodes, le concepteur C crée une méthode "lister\_étudiants" de façon à avoir comme description "Liste de tous étudiants ayant". L'ingénieur attribue à la méthode "lister\_étudiants" un code de retour de type "liste" ayant comme description "la liste des noms et adresses". Il attribue aussi à cette méthode, au moyen des boîtes de dialogue 25 et 26 illustrées dans les figures 5 et 6, les trois paramètres tels que définis dans le tableau A ci-dessous.

-21-

nom	description	type	valeur par défaut
"âge <"	"un âge inférieur à"	entier	26 ans
"lieu d'étude"	", étudiant à"	Paris, Versailles	Paris
"emprunt >"	"et ayant fait un emprunt supérieur à"	entier	6000 francs

TABLEAU A

5 En ce qui concerne la classe "impression", l'ingénieur C crée une méthode "imprime\_invitation" de façon à avoir une description telle que "Impression de lettres à :", et attribue à cette méthode un code de retour du type "void" signifiant que la méthode ne retourne pas de valeur, et deux paramètres tels que définis dans le tableau B ci-dessous.

10

nom	description	type	valeur par défaut
"destinataire"	"Impression de lettres à"	liste	lister_étudiants de consultation_système
"type de lettre"	"type de lettre :"	invitation d'anniversaire rappel information sur l'emprunt Étude	information

TABLEAU B

15 Ces tableaux indiquent comment remplir les boîtes de dialogue 25 et 26 des figures 5 et 6 pour constituer les deux classes "consultation\_système" et "impression" à l'aide de la boîte de dialogue 23 de la figure 3. Plus généralement, l'interface documentaire d'un objet est créée à partir des descriptions correspondant à cet objet, ses attributs, ses méthodes et les paramètres des méthodes.

20

Pour former une interface documentaire, le concepteur active dans la fenêtre 22 de la figure 2 le menu "Exporter à U". Cette commande appelle le générateur 14 pour générer une interface documentaire de l'objet sélectionné. Le générateur 14 extrait les descriptions de l'objet et les organise  
5 pour former l'interface documentaire.

La figure 11 illustre l'interface documentaire 18a de la méthode "lister\_étudiants". On y voit que l'interface documentaire 18a a été obtenue par extraction des descriptions de la méthode et de ses paramètres, telles que  
10 définies dans le tableau A. En particulier, les valeurs "26 ans", "Paris" et "6000 francs" sont les valeurs par défaut indiquées au tableau A. De manière générale, une interface documentaire 18 comprend du texte et au moins un champ possible 19 dont le contenu initial est fait de valeurs par défaut et peut être modifié par l'utilisateur U.

La figure 12 illustre un exemple d'interface documentaire 18b de la méthode "imprime\_invitation" de la classe "imprimer" définie précédemment. Les exemples des figures 11 et 12 suffisent à l'homme du métier pour savoir comment construire une interface documentaire relative à  
20 tout un objet. D'une manière générale, lorsque l'objet contient toutes les descriptions, elles sont de préférence organisées par le générateur 14 dans l'ordre suivant :

1. la description de l'objet (non illustrée), par exemple "impression" ;

25 2. la description de chaque attribut (non illustrée) , par exemple le numéro de l'imprimante, la qualité d'impression, la couleur de l'encre, le papier d'impression ; cette description est suivie d'un champ qui correspond à sa valeur ; si l'attribut a un nombre limité de valeurs, ce champ est une liste contenant toutes les valeurs possibles et ne faisant apparaître que la valeur  
30 sélectionnée ;



3. la description de chaque méthode (voir tableaux A et B, figures 11 et 12 et le texte s'y rapportant), cette description étant attachée et de préférence suivie de :

3.1. la description de sa valeur de retour, attachée ou suivie d'un champ qui représente cette valeur, et

3.2. la description de chacun de ses paramètres, attachée ou suivie d'un champ (19) représentatif de la valeur du paramètre ;

4. un moyen de commande (voir figure 12) indiquant quand la méthode doit être exécutée, l'exécution pouvant être immédiate ou différée, soit à un moment déterminé par une date et une heure, soit intervenir à la suite d'une autre méthode définie.

4.1. Dans le premier cas, un bouton de commande est libellé "Exécution immédiate" ;

4.2. Dans le second cas, un bouton de commande inclut une étiquette de valeur "à" (par exemple, --imprimer "à"--) suivie d'un champ "heure" et d'un champ "date" lié à un calendrier des jours ouvrables de l'année;

4.3. Dans le troisième cas, un bouton de commande inclut une étiquette intitulée "après la fin d'exécution de" suivie d'un champ permettant de choisir une méthode d'un objet quelconque et signifie que la méthode sera exécutée après la fin d'exécution de la méthode sélectionnée.

Le concepteur C a la possibilité de changer la présentation de la page, ajouter ou enlever des champs, sélectionner les descriptions à prendre en compte, et modifier le texte de descriptions et des types pour rendre la compréhension du document plus facile. Il s'ensuit qu'un même objet peut générer plusieurs interfaces documentaires adaptées à des besoins particuliers offerts dans l'objet. Par exemple, on a vu précédemment que l'objet "impression" peut avoir un grand nombre d'attributs et de méthodes, qui offre un grand choix d'interfaces documentaires en fonction des attributs

et méthodes adaptées aux besoins. Il est bien sûr conseillé au concepteur C de faire les interfaces documentaires 18 avec l'aide de l'utilisateur U.

La description va maintenant porter sur le module de commande  
5 12 servant à l'utilisateur U pour obtenir les interfaces documentaires 18 de la base 15. Le module 12 contient un bloc 40 de génération de documents 41 issus des interfaces documentaires stockés dans la base 15. Le bloc 40 et donc connecté à la base 15 pour y prélever les interfaces documentaires 18 et déterminer les documents de commande 41 adaptés aux besoins particuliers  
10 de l'utilisateur U. Par exemple, en supposant que l'utilisateur a besoin du document "impression", une interface incluant la méthode illustrée dans la figure 12 va se présenter à l'utilisateur, qui pourra modifier les valeurs des champs 19 et sélectionner les boutons de commande d'exécution pour en faire un document 41. Le document 41 illustré dans la figure 12 correspond à  
15 l'interface documentaire 18 dans laquelle les valeurs des champs 19 ont maintenues inchangées par l'utilisateur et le bouton d'exécution différée à une heure prédéterminée a été activé (en grisé dans la figure 12).

La figure 13 illustre un exemple de boîte de dialogue 42  
20 présentée à l'utilisateur U par le module d'utilisateur 12. La barre de menus de la boîte 42 contient notamment deux menus principaux "Documents" et "Politiques". Dans le menu "Documents" se trouvent la commande "Rechercher" pour trouver des interfaces documentaires, par exemple à partir des descriptions des objets correspondants, et la commande "Ouvrir" pour  
25 lister les interfaces documentaires par le nom de leurs objets correspondants et y sélectionner un nom d'objet. Un objet sélectionné s'affiche avec la description de l'objet. De cette interface, l'utilisateur en fait le document désiré. Dans le menu "Documents" se trouvent aussi bien sûr les commandes de sauvegarde (Enregistrer et Enregistrer sous...) et de fermeture du  
30 document.

-25-

Le menu "Politiques" contient notamment deux commandes "Afficher" et "Afficher l'historique". La commande "Afficher" (display) affiche l'ensemble des documents créés par l'utilisateur dans l'ordre d'exécution des commandes qu'il a choisi. Cet ensemble définit la politique de l'utilisateur. Il dispose ainsi d'une vue globale des méthodes qu'il a activées à partir de plusieurs documents. Il peut la valider ou la modifier. Par exemple, le fait de cliquer deux fois sur une méthode choisie entraîne l'ouverture du document correspondant pour d'éventuelles modifications. La commande "Afficher l'historique" présente toutes les méthodes qui ont été exécutées par l'utilisateur, l'état de fin d'exécution, et l'heure. Le menu contient aussi les commandes de sauvegarde et de fermeture.

De manière générale, la description qui précède fait ressortir un procédé de génération d'une interface 18 pour la commande d'un système informatique 10 par un utilisateur U, la commande étant faite à partir d'au moins un objet incluant des descriptions de l'objet et/ou d'au moins un membre de l'objet, le procédé consistant à extraire de l'objet au moins en partie lesdites descriptions et à les organiser pour traduire le sens de ladite commande en un langage compréhensible à l'utilisateur et ainsi faire de ladite interface une interface documentaire.

L'outil permet au concepteur C de facilement adapter le texte des descriptions extraites, de façon que leur organisation rende compréhensible le texte de l'interface. Pour rendre plus souple le procédé, le concepteur pourrait sélectionner les descriptions à extraire.

De préférence, les descriptions incluant au moins une description d'une méthode de l'objet, le procédé consiste à attacher à cette description une description de son code de retour et/ou une description d'au moins un paramètre respectif de la méthode et une valeur 19 du paramètre dont la valeur peut être modifiable. Dans ce cas également et comme dans l'exemple illustré, le procédé peut consister avantageusement à ajouter à l'interface 18

un moyen de commande, par l'utilisateur, du moment d'exécution de ladite interface.

Dans l'exemple décrit, le texte des descriptions servant à faire  
5 l'interface est fait par un concepteur utilisant le système, en réponse à des  
besoins qui lui sont soumis par l'utilisateur. Cette coopération entre le  
concepteur et l'utilisateur rend le procédé très souple, très évolutif et très  
performant. Cependant, il est évident que cette condition n'est pas nécessaire,  
l'outil étant par exemple adapté à un métier donné ou une fonction  
10 d'entreprise donnée.

Enfin, on a vu précédemment l'avantage que pouvait tirer  
l'utilisateur de la possibilité offerte dans l'exemple décrit de sélectionner des  
interfaces et de définir leur ordre d'exécution temporel, et de lui présenter  
15 dans une fenêtre les interfaces sélectionnées et disposées dans ledit ordre.

Le module de conception 13 dispose en outre d'un outil de  
commande 50 de fonctions analogues dans différents systèmes d'exploitation  
20 et/ou applications et d'un outil de génération automatique de code 60 pour  
mettre en œuvre l'outil d'interface 50. Le système informatique 10 représenté  
sur la figure 1 sera considéré comme un système très hétérogène,  
représentatif d'un cas difficile à gérer. Dans l'exemple considéré, les deux  
systèmes d'exploitation 4a et 4b étaient deux versions différentes du système  
25 d'exploitation UNIX, connues sous les noms UNIX Berkeley et UNIX AT&T.  
Les deux machines 2 sont aussi supposées inclure dans leurs mémoires  
respectives 3 deux logiciels d'impression différents l'un de l'autre 6a et 6b,  
tels que les logiciels connus sous leurs noms de marques déposées DPF  
OpenSpool et XPRINT.

30

On suppose que les deux systèmes d'exploitation 4a, 4b et les  
deux logiciels d'impression 6a, 6b ont au moins une fonctionnalité commune

F. Cependant, les logiciels 4a, 4b, 6a, 6b pouvant être très différents l'un de l'autre, cette fonctionnalité F peut être exécutable selon diverses commandes différentes d'un logiciel à un autre.

5           Plusieurs fonctionnalités peuvent être communes à tout ou partie des logiciels installés, tels que l'impression de documents et la sauvegarde de documents. La description portera sur l'impression de documents comme exemple de fonctionnalité F. Cette fonctionnalité peut comprendre plusieurs fonctions f, par exemple les fonctions : f1 "imprimer" ;  
10 f2 "lister les impressions en cours" ; f3 "visualiser les files d'attente" ; f4 "purger les impressions" ; et f5 "changer la priorité d'une impression dans une file". Une fonctionnalité F est donc dans ce cas une famille de fonctions f. L'exemple qui suit portera sur la fonction f1 et suffira à l'homme du métier pour comprendre l'invention dans sa généralité.

15           On suppose que la fonction f1 "imprimer" dans les quatre logiciels 4a, 4b, 6a et 6b dans le système informatique 10 de la figure 1 est faite par quatre commandes spécifiques différentes respectives : Pa et Pb pour les systèmes d'exploitation respectifs 4a et 4b ; et Pc et Pd pour les logiciels  
20 d'impression 6a et 6b. Dans l'exemple des logiciels considérés, les deux versions différentes 4a et 4b du système d'exploitation UNIX ont des commandes d'impression respectives Pa = "lpr" et Pb = "lp". Néanmoins, on pourrait aussi avoir des commandes d'impression telles que : "enq" pour le système d'exploitation connu sous le nom de marque déposée AIX ; "print"  
25 pour le système d'exploitation DOS pour ordinateur personnel ; "np" pour le logiciel d'impression DPF OpenSpool ; et "xpad" pour le logiciel d'impression connu sous le nom de marque déposée XPRINT.

          Chaque commande spécifique Pa-Pd comprend des options  
30 éventuellement associées à des types particuliers respectifs. Le tableau C suivant illustre un exemple d'un inventaire concernant seulement les

commandes d'impression "lpr" et "lp" des systèmes d'exploitation 4a et 4b. Cet exemple suffit à l'homme du métier pour comprendre l'invention.

Options de "lpr"	suivies de	Options de "lp"	suivies de	options - commentaires
-P	chaîne	-d	chaîne	destination de l'impression
-T	chaîne	-t	chaîne	titre du document
-#	num	-n	num	nombre de copies
n'existe pas		-w		écriture d'un message sur terminal de C à la fin de l'impression
-r		n'existe pas		destruction du fichier lorsque l'impression est lancée

5

TABLEAU C

L'outil de commande 50 est un outil logiciel ayant pour but principal d'offrir à un utilisateur informaticien, le concepteur C dans la figure 1, une commande commune P0 pour l'exécution de chacune des commandes d'impression différentes Pa-Pd dans l'exemple illustré. Comme indiqué à la figure 1, l'outil illustré 50 comprend un bloc 51 de définition de la commande commune P0, et deux pilotes d'impression (drivers) 52a, 52b pour les deux machines respectives 2a et 2b, les pilotes réagissant à la commande commune P0 pour générer la commande Pa ou Pc dans la machine 2a et Pb ou Pd dans la machine 2b. Dans l'exemple illustré, les deux pilotes 52a et 52b sont implantés dans les deux mémoires 3 des deux machines 2 tandis que le bloc 51 est incorporé au module de conception 13. Le bloc de définition 51 est connecté aux deux pilotes.

20

L'outil de commande 50 met en œuvre un procédé de commande d'une fonction f exécutable sous des commandes différentes Pa-Pd dans des logiciels respectifs 4a, 4b, 6a, 6b, les commandes pouvant avoir chacune au

-29-

moins une option. Le procédé consiste à définir dans une classe abstraite une méthode abstraite pour la fonction f, la méthode abstraite incluant des paramètres correspondant à l'union, au sens logique, de toutes les options (tableau C) des commandes spécifiques, à définir une commande commune (P0) incluant des symboles arbitraires correspondant aux paramètres de la méthode abstraite, à former au moins un pilote (52) pour implémenter la méthode abstraite dans une machine, et à faire exécuter par le pilote l'une des commandes spécifiques avec des options équivalant aux options de la commande commune.

10

Supposons que l'utilisateur U désire exécuter la commande : imprimer en deux exemplaires, avec une imprimante appelée imprimante 1 du système informatique 10, le fichier appelé *monfichier*, en indiquant en en-tête le nom du fichier, et détruire le fichier une fois imprimé. Cette commande dans les logiciels 41 et 4b s'écrit de la façon suivante, compte tenu du tableau C :

```
lpr      -Pimprimantel -#2 -r monfichier
lp       -dimprimantel -n2 monfichier
```

où il est à noter du tableau C que la suppression du fichier n'est pas prévue pour la commande "lp".

Selon le procédé précité, on forme une méthode abstraite pour la fonction f. Une méthode générale comprenant un en-tête et un corps, une méthode abstraite n'a qu'un en-tête, de sorte qu'elle n'est pas exécutable et que la classe correspondante ne peut donc pas être instanciée. Dans l'exemple de la fonction f1 "imprimer", on crée donc une méthode abstraite que l'on appellera du même nom "imprimer". Cette méthode inclut des paramètres correspondant respectivement aux options des commandes spécifiques Pa-Pd, le nombre de paramètres correspondant au nombre formé par l'union, au sens logique (assembliste), de toutes les options des commandes. Ainsi, dans l'exemple de la fonction "imprimer", et en illustrant les exemples dans le

-30-

langage Java®, il résulte du tableau C que la méthode abstraite "imprimer" est définie par les six paramètres suivants :

```
public static int imprimer
5      (      String destination,
          String titre,
          int nombreDeCopies,
          boolean message,
          boolean destruction,
10     String nomDuFichier
      ) ;
```

Des six paramètres ci-dessus, les cinq premiers correspondent aux cinq options définies dans le tableau C et le dernier paramètre désigne le  
15 nom du fichier qui se place à la fin des commandes d'impression "lpr" et "lp".

Une classe abstraite, au sens orienté objet, contient au moins une méthode abstraite, ici la méthode abstraite "imprimer" définie pour l'ensemble des commandes analogues Pa-Pd de la fonction donnée f1. La  
20 classe abstraite pourrait être quelconque. De préférence, comme dans l'exemple considéré, la classe abstraite correspond à la fonctionnalité F et a, dans l'exemple illustré, le même nom "impression". La classe abstraite "impression" n'a donc comme méthodes que toutes les méthodes abstraites se rapportant aux fonctions f1-f5 de la fonctionnalité F. La classe "impression"  
25 correspond à une interface, telle qu'elle est définie par exemple dans le langage Java®. Une telle interface est définie comme la classe la plus abstraite qu'on puisse construire, c'est-à-dire une classe dont chaque méthode est abstraite.

30 On a vu que le procédé mis en œuvre par l'outil 50 consiste d'autre part à définir une commande commune P0 incluant des symboles



-31-

arbitraires correspondant aux paramètres de la méthode abstraite. Par exemple, la commande commune P0 a la syntaxe suivante :

```
imprimer [ -dDestination] [ -tTitre ] [ -nNombre] [-m] [-D]
5 -ffichier
```

ayant comme symboles arbitraires -d, -t, -n, -m, -D et -f. Ces symboles peuvent ou non être associés à des options, celles-ci étant mises entre crochets, de sorte que seul le terme "fichier" n'est pas une option, comme cela  
10 a été indiqué précédemment. L'absence d'option signifiant que le symbole correspond à un type booléen.

Le procédé tel que défini précédemment consiste aussi à construire les pilotes 52a et 52b pour générer les commandes Pa et Pb ("lpr"  
15 et "lp") dans les deux machines 2a et 2b respectives. Un pilote implémente, dans le sens de la technologie orientée objet, l'interface "impression", c'est-à-dire qu'un pilote définit un corps à chaque méthode de l'interface, de façon que la classe abstraite puisse générer une classe exécutable. Le corps ainsi défini permet de construire les options de la commande correspondante ("lpr"  
20 par exemple) avec les paramètres de la méthode et de lancer cette commande.

L'exemple qui suit illustre comment créer le pilote 52a relatif à la commande "lpr" et indique clairement à l'homme du métier comment d'autres pilotes peuvent être créés. Le pilote 52a est une classe qui permet  
25 d'implémenter l'interface impression de la façon suivante.

```
public class Pilote_lpr implements Impression
{
    public static int imprimer
    (
        String destination,
30      String titre,
        int nombreDeCopies,
        boolean message,
```

-32-

```
        boolean destruction,  
        String nomDuFichier  
    )  
    {  
        // 1) construction des arguments de lpr à partir des  
5         paramêtres.  
        // 2) lancement de la commande lpr avec ses arguments.  
        // 3) récupération du code de retour et des messages  
            de lpr et traitement de ceux-ci.  
    }  
  
10  
  
    public static void main (String argv [])  
    {  
        // le premier argument de main est le nom de la  
        // méthode à exécuter, les autres sont les arguments  
        // de cette méthode.  
15        // En fonction du nom de la méthode, on transforme les  
        // arguments de main dans le type des paramêtres de la  
        // méthode que l'on appelle ensuite  
    }  
}  
  
20
```

Plus généralement, d'autres possibilités de pilotes existent, par exemple un ou les deux pilotes pourraient être installés aussi dans l'interface de commande 11, ou un seul pilote suffirait, avantageusement installé dans l'interface de commande 11. De même, il peut être prévu seulement un seul  
25 pilote pour un système n'ayant qu'une seule machine, le pilote servant à générer les diverses commandes des logiciels installés dans cette machine.

Enfin, le procédé défini précédemment consiste à faire exécuter par le pilote l'une des commandes spécifiques, par exemple "lpr", avec des  
30 options équivalant aux options de la commande commune.

L'équivalence entre les options de la commande "lpr" et les options de la commande commune P0 peut se faire de diverses façons. La

-33-

façon préférée qui va être illustrée consiste à créer un fichier de configuration définissant les types et les valeurs par défaut des options de chaque commande spécifique pouvant être exécutée par le pilote. Dans le présent exemple, la commande spécifique est "lpr", qui sera exécutée par le pilote 52a.

5 De préférence, le fichier de configuration comprend en outre le type de chaque option et/ou des commentaires éventuels et/ou le nom du pilote.

Le fichier de configuration est de préférence créé préalablement à la construction et avantageusement lors de l'installation du pilote. Dans l'exemple choisi, le fichier de configuration fourni par le pilote est appelé "ConfigImpression". Tout autre nom peut lui être donné, mais il importe que ce nom soit le même pour tous les pilotes de la même fonction f des commandes concernées Pa-Pd et qu'il soit accessible par tous ces pilotes, de préférence en étant installé dans un même répertoire..

15 Le fichier de configuration de notre exemple est :

```
NomDuPilote=Pilote_lpr
OPTION=-d TYPE=chaîne DEFAULT=null
20 // le type de la valeur associée à l'option -d est une
// chaîne de caractères
OPTION=-t TYPE=chaîne DEFAULT=null
// le type de la valeur associée à l'option -t est une
// chaîne de caractères
25 OPTION=-n TYPE=num DEFAULT=1
// le type de la valeur associée à l'option -n est un nombre
// entier
OPTION=-D TYPE=none
// il n'y a pas de valeur associée à l'option -D
30 OPTION=-f TYPE=chaîne OBLIGATOIRE
// l'option -f doit être présente dans la ligne des
// commandes
// L'option -m n'existe pas pour ce pilote
```

On retrouve dans le fichier de configuration les symboles utilisés dans la commande commune P0.

5           L'équivalence entre options de la commande commune et de la commande spécifique consiste en outre à déterminer les paramètres de la commande spécifique par consultation du fichier de configuration par la commande commune. Ainsi, la commande commune P0 lit le fichier de configuration afin de vérifier que les options de la syntaxe de la commande  
10 commune sont bien compatibles avec celles du pilote installé. En d'autres termes, la commande commune P0 détermine si les options fournies par le concepteur C sont légales. Si oui, elle est complétée par l'ensemble des options non fournies et elle appelle le pilote pour exécution de la commande spécifique. Sinon, un message d'erreur est émis avec la description des  
15 options légales. D'autres part, elle vérifie si toutes valeurs attachées aux options sont bien affectées. Si la valeur due à une option n'existe pas, elle est déterminée à partir de sa description (valeur par défaut) dans le fichier de configuration.

20           Sachant que la commande d'impression "lpr" telle que définie précédemment est :

```
lpr           -Pimprimantel -#2 -r monfichier
```

25           alors la commande commune P0 correspondante est :

```
imprimer -n2 -dImprimantel -D -fmonfichier
```

30           La commande commune P0 appelle alors le pilote 52a relatif à la commande "lpr" de la façon qui suit :

-35-

```
java PiloteLpr imprimer Imprimantel null 2 false true  
monfichier.  
// false indique que l'option -m n'est pas positionnée  
// true indique que l'option -D l'est  
5 // null indique que l'option -t n'est pas renseignée
```

Cette commande provoque l'appel de la méthode principale "main" du pilote 52a qui, à son tour, déclenche l'appel de la méthode "imprimer" de la façon suivante:

```
10  
imprimer («Imprimantel», null, 2, false, true, «monfichier»)  
;
```

La méthode "imprimer" lance ainsi la commande "lpr" sous la forme :

```
15  
lpr -Pimprimantel -#2 -r monfichier
```

Jusqu'à présent, on a décrit la fonction d'un pilote d'application.

20 On a vu qu'un pilote est une classe servant à l'implémentation d'une méthode abstraite d'une classe abstraite. Le pilote est défini par des données écrites par le concepteur C à l'aide de l'outil 20 ou 30 du module de conception 13. Les lignes de code de ce logiciel doivent normalement être entièrement écrites par le concepteur C. Une grande partie de ce travail est très fastidieux et donc

25 coûteux en temps et en argent. Cependant, la génération automatique pose deux problèmes. Le premier problème est de traduire le pilote en un langage orienté objet. Le second problème est de générer automatiquement un corps à la méthode abstraite. On a vu que le corps de la méthode abstraite doit inclure les données spécifiques de la commande spécifique à exécuter, et un

30 pilote peut servir à l'exécution de plusieurs commandes spécifiques. Ceci ne peut plus être fait au moyen des outils 20 et 30 et requiert des opérations manuelles du concepteur C. Les difficultés soulevées par ces deux problèmes amènent le concepteur à écrire les pilotes.

Le système informatique 10 de l'invention inclut un outil 60 de génération automatique, en un langage orienté objet, d'un pilote 52 d'implémentation d'une méthode abstraite d'une classe abstraite pour l'exécution d'une commande spécifique Pa-Pd ayant des données spécifiques (tableau C), la méthode abstraite ayant une valeur de retour et au moins un paramètre pourvu d'un type. Le langage orienté objet choisi dans l'exemple qui suit est le langage Java®. L'outil 60 de génération automatique d'un pilote qui va maintenant être décrit est contenu dans le module de conception 13, bien qu'il pourrait être inclus ailleurs dans le système informatique.

Le procédé mis en œuvre par l'outil 60 consiste à définir la classe du pilote par des données de définition et en faisant correspondre ladite valeur de retour et ledit type dudit paramètre à des types correspondants dans ledit langage, à traduire les données de définition du pilote dans ledit langage, à générer le corps de la méthode abstraite en construisant un squelette à compléter par le concepteur C avec les données spécifiques de la commande spécifique.

Pour la définition du pilote, le concepteur C utilise de préférence l'outil 20 ou 30. Par exemple, il peut utiliser un état vierge de la boîte de la figure 3 pour indiquer le nom de la classe, ajouter au moins un attribut à l'aide de la boîte de la figure 4, et au moins une méthode et au moins un paramètre avec les boîtes des figures 5 et 6. Cependant, pour faire la génération automatique, l'outil 60 nécessite une correspondance entre les types des paramètres et de la valeur de retour de la méthode et des types dans le langage choisi, ici en langage Java. Sans cette correspondance, le concepteur devrait écrire manuellement le logiciel du pilote. La correspondance choisie à titre d'exemple se traduit par la règle suivante en langage Java® :

-37-

« Tout type défini par le concepteur C doit appartenir à l'ensemble (T) suivant :

{Boolean, Character, Double, Float, Integer, Long, String, Void}

5                   ou

List(x)

avec x appartenant à (T). »

L'ensemble T peut être étendu à List (List(x)) et successivement  
10 pour représenter des tableaux à plusieurs dimensions. De manière générale, l'ensemble T est un tableau pouvant être multidimensionnel. Il est évident qu'on peut remplacer cet ensemble (T) par un autre ensemble avec lequel il est en bijection. Par exemple, en France le type défini en anglais par Character est remplacé par Caractère.

15

Chaque élément de l'ensemble T correspond à une classe prédéfinie de même nom du langage Java, sauf List et Void, qui correspondent respectivement à la classe Vector et au mot réservé void de Java®. En d'autres termes, les types de l'ensemble T incluent de préférence  
20 des classes prédéfinies de même nom du langage.

La figure 14 illustre une boîte de dialogue 61 s'ouvrant pour définir les types de paramètres et la valeur de retour de la méthode. Si par exemple le concepteur veut définir une méthode libellée "copier" qui ne  
25 retourne pas de valeur et qui a les deux paramètres suivants :

- 1) "nom du fichier" (de type String) qui est l'adresse URL (Uniform Resource Locator) du fichier, et
- 2) "répertoire" (de type String) qui est l'adresse URL du répertoire dans lequel on veut copier le fichier,

30 il indique le nom "copier" dans le champ texte définissant le nom de la méthode dans la boîte 25 de la figure 5, il sélectionne dans la boîte 61 le type void pour l'introduire dans la champ texte relatif au type de retour de la

-38-

méthode et actionne le bouton de commande "Ajouter" pour définir un nouveau paramètre avec l'aide de la boîte 26 de la figure 6. Dans la boîte 26, il indique le nom du premier paramètre "nom du fichier" et sélectionne dans la boîte 61 le type String, qui vient se placer dans le champ texte relatif au type du paramètre. Le concepteur ajoute de préférence, dans la zone de texte relative à la description du paramètre, le texte par exemple "adresse URL du fichier". Il en est de même pour le second paramètre.

La méthode abstraite "copier" s'écrit donc :

10

```
public void copier(String nom_du_fichier , String  
répertoire) ;
```

On supposera que le nom de la classe abstraite correspondante, une interface au sens de Java® dans l'exemple illustré, se nomme "opérations sur fichier".

On a vu qu'une autre étape du procédé consiste à traduire les données de définition du pilote dans le langage choisi. La transcription peut se faire selon l'exemple suivant. Le nom de l'interface est le nom de la classe définie par le concepteur C, précédé des deux caractères « I\_ ». Si le nom de cette classe contient le caractère blanc « », qui est interdit dans les langages classiques comme Java ou C, on le remplace par « \_ ». On applique la même règle pour la classe du pilote, mais en remplaçant « I\_ » par « P\_ ». Si on génère un nombre n de pilotes, on fait suivre les noms des pilotes par les deux caractères « \_i » où i varie de 1 à n. Si les noms comportent des lettres accentuées, on peut aussi prévoir la suppression des accents pour convenir au langage Java ordinaire. Ainsi, la classe s'écrit de la façon suivante :

```
30 interface I_operations-sur_fichier  
{
```



-39-

```
public void copier(String nom_du_fichier , String
repertoire) ;
}
```

5 et le pilote correspondant s'écrit de la façon suivante :

```
class P_operations-sur_fichier_1 implements
I_operations_sur_fichier
{
10 public void copier(String nom_du_fichier , String
repertoire) ;
{
...
}
15 public static void main (String argv[])
{...}
}
```

où

- 20 - le contenu {...} relatif à la méthode "copier" constitue le corps de la méthode, qui correspond à la commande spécifique à exécuter, et
- main est la méthode principale qui va être appelée par la commande unifiée P0, qui déclenche à son tour l'exécution de la méthode "copier" pour l'exécution de la méthode spécifique "lpr". Il est à noter ici que l'exécution du
- 25 pilote peut être faite de plusieurs façons connues de l'homme du métier. L'exemple d'exécution choisi consiste à incorporer au pilote la méthode principale main.

Si par exemple la classe "impression" définie précédemment par

30 le concepteur C contient un attribut "nombre de copies" de type Integer et ayant comme valeur par défaut "1", la classe en langage Java® du pilote a pour attribut :

-40-

```
Integer Nombre_de_copies = new Integer (1) ;
```

On rappelle que c'est l'utilisateur U qui, au moyen d'un document 41, va demander l'exécution de la commande commune P0 par l'intermédiaire de l'interface documentaire 18 correspondante. La commande commune P0 va appeler le pilote par l'intermédiaire de la commande suivante :

```
java P_Impression_1 imprimer 2 monfichier monimprimante
```

La méthode principale main du pilote va alors être déclenchée. La méthode principale main a, de façon bien connue, tous ses paramètres de type String. La méthode principale main va appeler la méthode "imprimer" qui a des paramètres variés, tels que :

```
imprimer(Integer p1, String p2, String p3)
```

Chaque type de paramètre dans la méthode main est donc transformé en le type du paramètre correspondant de la méthode "imprimer". Ceci peut se faire automatiquement dans la méthode

```
main (String argv []){...}
```

où le premier argument est le nom de la méthode implémentée de l'interface qu'il faut appeler. Le (i+1)<sup>e</sup> argument de la méthode main correspond au i<sup>e</sup> argument de la méthode appelée. Son type doit être cependant changé si l'argument de la méthode n'est pas du type String. Dans le dernier exemple, la méthode main va appeler la méthode "imprimer" du pilote de la manière suivante :

```
imprimer(new Integer(argv[1]), argv[2], argv[3]) ;
```

-41-

Si `argv[1]` ne correspond pas à un entier, une exception sera levée et retransmise dans le module de commande 12.

On va maintenant décrire un exemple de génération du corps de la méthode "imprimer". On rappelle que c'est dans le corps de la méthode "imprimer" que se fait l'appel à la commande spécifique Pa ("lpr"). Tout un mécanisme en langage Java permet au pilote d'exécuter la commande spécifique et de recevoir ses résultats. Ce mécanisme est décrit dans l'exemple du paragraphe suivant.

10

L'interface documentaire 18 de l'utilisateur U appelle commande commune P0, qui elle même appelle le pilote 52a, après consultation de son fichier de configuration, de la façon suivante :

15 `java nomDuPilote nomDeLaMéthode paramètresDeLaMéthode`

Une étape du procédé consiste à générer automatiquement le corps de la méthode abstraite en construisant un squelette à compléter par le concepteur C avec les données spécifiques de la commande spécifique "lpr". Le squelette se fait par exemple par construction de chaînes. Dans notre exemple, la construction du squelette à compléter se fait de la manière suivante :

```
25 // ##### début de modification
String commandStr = "##### le nom de la commande" + " " ;
commandStr = commandStr+ " " + "##### l'option"+
pl.toString() ;
commandStr = commandStr+ " " + "##### l'option"+ p2 ;
commandStr = commandStr+ " " + "##### l'option"+ p3 ;
30 // ##### fin de modification
```

Pour compléter le squelette, il est de préférence indiqué où les données sont à compléter et/ou quelles données sont à compléter par le concepteur C. Dans l'exemple précédent, le concepteur n'a qu'à chercher dans la classe du pilote l'occurrence des caractères successifs ##### . Quand il

5 trouve cette occurrence, le squelette lui indique où et quelles données il doit y introduire pour le compléter. Suivant l'exemple précédent, il va corriger le code de la manière suivante :

```
// début de modification
10 String commandStr = "lpr";
   commandStr = commandStr+ " " + "-#" + p1.toString() ;
   commandStr = commandStr+ " " + "-P" + p2 ;
   commandStr = commandStr+ " " + p3 ;
   // fin de modification
15
   où   commandStr   va prendre comme valeur possible

   "lpr -#2 -Pimprimantel monfichier".
```

20 L'appel de cette commande sera ensuite automatique.

On va maintenant décrire un exemple de mise en oeuvre de l'outil 60 pour la formation d'un squelette de pilote généré en langage Java et destiné à une machine fonctionnant sous le système d'exploitation connu sous

25 le nom de marque déposée Windows NT.

L'exemple part d'une classe simple qui contient une seule méthode, la généralisation étant évidente. Le nom de la classe est par exemple "copie de fichier". La méthode de cette classe est "copie", qui retourne

30 la valeur void et qui a les deux paramètres suivants :

- "fichier", de type String, et
- "répertoire de destination", de type String.

Le concepteur ouvre la fenêtre 21 de la figure 2 et actionne par exemple une\_commande "génération de pilote" (non illustrée), se trouvant de préférence dans le menu "Exporter à U". Cette commande génère le code

5 suivant :

```
interface I_Copie_de_fichier
{
    void copie (String fichier ; String repertoire_de_destination ) ;
}

10
et

public class I_Copie_de_fichier_1 implements I_Copie_de_fichier
{
    void copie (String fichier ; String repertoire_de_destination )
15
    {
        try
        {
            Process p ;
            Runtime rt = Runtime.getRuntime () ;

            // ##### début de modification
20
            String commandStr = "##### le nom de la commande";
            commandStr = commandStr+ " " + "##### l'option"+
            fichier;
            commandStr = commandStr+ " " + "##### l'option"
                                + repertoire_de_destination ;
25
            // ##### fin de modification

            p = rt.exec("C:\\WINNT\\System32\\cmd /c "+
            commandStr) ;

30
            BufferedReader br =
                new BufferedReader(
                    new InputStreamReader(
                        p.getInputStream())) ;

35
            String result;
            while (( result = br.readLine()) != null )
            {
                System.out.println("~~~~~ " + result) ;
                StringTokenizer St =
                    new StringTokenizer ( result ) ;
```

-44-

```

/*
while ( St.hasMoreTokens() )
{
    st = St.nextToken();
    System.out.println("hello\n");
5    System.out.println
        ("token = " + st + ".");
    }
*/

}

10    }
    catch ( IOException ie )
    {
        System.out.println("IO Exception") ;
    }
    catch ( SecurityException se )
15    {
        System.out.println ("No process");
    }
}

public static void main (String argv [])
20    {
        argv[0](argv[1] , argv[2]) ;
    }
}

```

Dans ce code, le mot "br" désigne le résultat de l'exécution de la  
 25 commande spécifique. Le concepteur C cherche alors l'occurrence de ##### et  
 change

```

// ##### début de modification
String commandStr = "##### le nom de la commande";
30    commandStr = commandStr+ " " + "##### l'option"+
        fichier;
    commandStr = commandStr+ " " + "##### l'option"
        + répertoire_de_destination ;
// ##### fin de modification
35
en

// début de modification
String commandStr = "copy";
40    commandStr = commandStr+ " " +fichier;

```

-45-

```
commandStr = commandStr+ " " +  
répertoire_de_destination ;  
// fin de modification
```

## Revendications

1. Procédé de commande d'une fonction (f) exécutable par divers logiciels (4a, 4b, 6a, 6b) sous des commandes (Pa-Pd) spécifiques aux logiciels respectifs et pouvant avoir chacune au moins une option, les logiciels étant installés dans au moins une machine (2a, 2b) d'un système informatique (10), caractérisé en ce qu'il consiste à définir dans une classe abstraite une méthode abstraite pour la fonction (f), la méthode abstraite incluant des paramètres correspondant à l'union, au sens logique, de toutes les options (tableau C) des commandes spécifiques, à définir une commande commune (P0) incluant des symboles arbitraires correspondant aux paramètres de la méthode abstraite, à former au moins un pilote (52) pour implémenter la méthode abstraite dans une machine, et à faire exécuter par le pilote l'une des commandes spécifiques avec des options équivalant aux options de la commande commune.

2. Procédé selon la revendication 1, caractérisé en ce que l'équivalence entre les options de la commande spécifique et les options de la commande commune consiste à créer un fichier de configuration (ConfigImpression) définissant les types et les valeurs par défaut des options de chaque commande spécifique pouvant être exécutée par le pilote, et à déterminer les paramètres de l'une de ces commande spécifiques par consultation du fichier de configuration par la commande commune.

3. Procédé selon la revendication 1 ou 2, caractérisé en ce qu'il consiste à faire correspondre un pilote (52a) à une machine (2a) du système informatique.

4. Procédé selon l'une des revendications 1 à 3, caractérisé en ce que la classe abstraite est la classe la plus abstraite pouvant être définie, telle qu'une interface dans le langage Java®.



5. Procédé selon l'une des revendications 1 à 4, caractérisé en ce que la classe abstraite contient tout ou partie des méthodes se rapportant à des fonctions d'une même fonctionnalité (F) commune aux logiciels.

5 6. Système informatique (10) comprenant au moins un machine disposant de logiciels divers (4a, 4b, 6a, 6b) ayant en commun au moins une fonction (f) exécutable sous des commandes spécifiques (Pa-Pd) aux logiciels respectifs et pouvant avoir chacune au moins une option, caractérisé en ce qu'il met en œuvre le procédé défini par l'une des revendications 1 à 5.

**THIS PAGE BLANK (USPTO)**

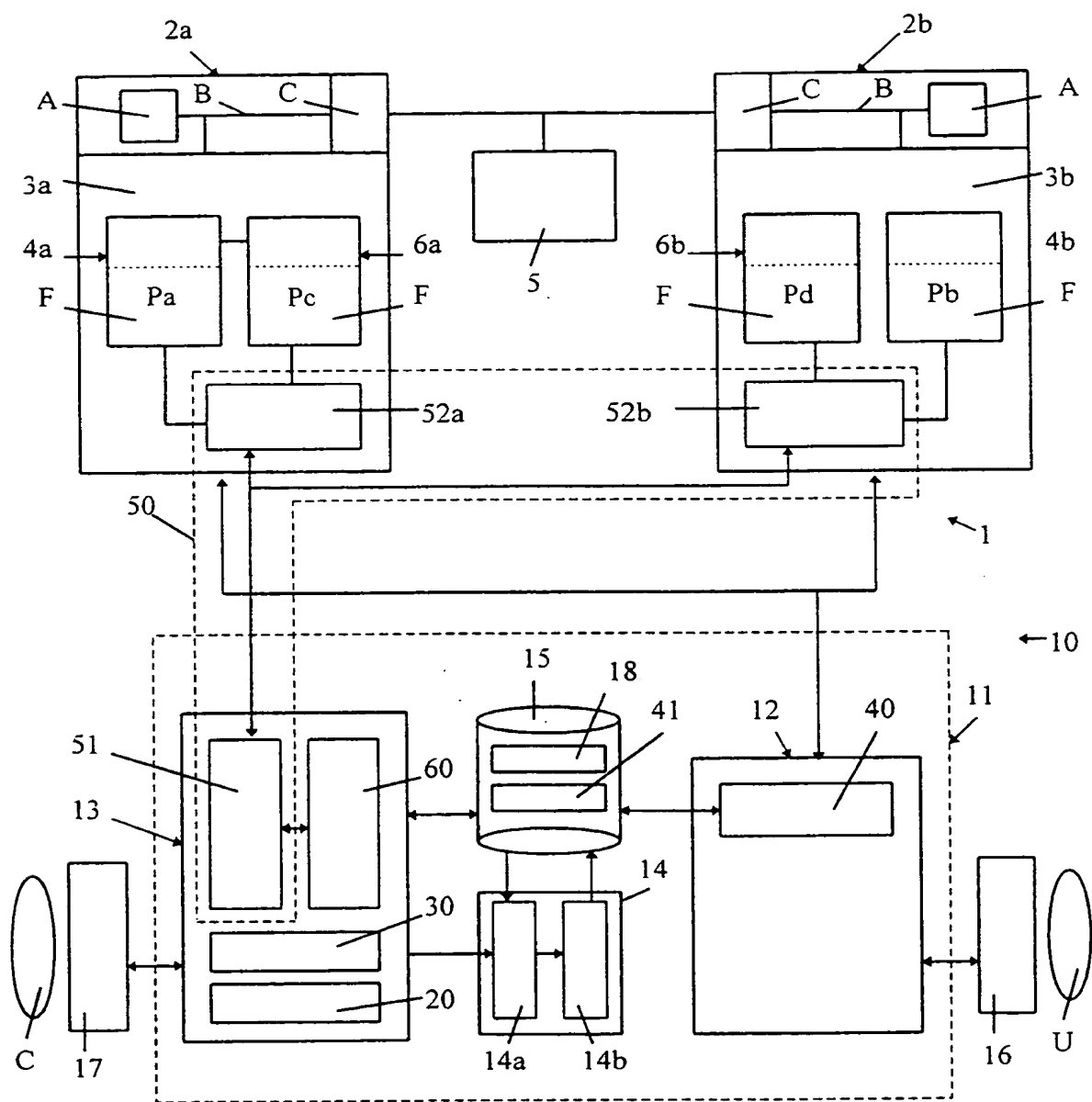


FIG. 1

**THIS PAGE BLANK (USPTO)**

20 22

Fichier	...	Classes	...	Rechercher	...	Objets	...	Exporter à U	Aide
		Nouvelle classe Nouvelle classe héritée Ouvrir une classe supprimer une classe Enregistrer la classe Enregistrer la classe sous... Fermer la classe				Nouvel objet de classe Cloner un objet Ouvrir un objet supprimer un objet Enregistrer l'objet Enregistrer l'objet sous... Fermer l'objet			

FIG. 2

21

Nom de la classe  Identification de la classe

Description de la classe 

Zone de texte

Attributs de la classe

Liste des attributs

Attribut1

Modifier l'attribut

Supprimer l'attribut

Ajouter un attribut

Méthodes de la classe

Liste des méthodes

Méthode1

Modifier la méthode

Supprimer la méthode

Ajouter une méthode

Classe

FIG. 3

23

20

**THIS PAGE BLANK (USPTO)**

Nom de l'attribut		Champ texte
Description de l'attribut	Valeurs possibles de l'attribut	
Zone de texte	Zone de texte	
Type de l'attribut	Champ texte	Valeur par défaut de l'attribut
		Champ texte
Valeur réelle de l'attribut		Champ texte

FIG.4

24

20

Nom de la méthode	Champ texte	Type de retour de la méthode	Champ texte
Description de la méthode	Paramètres de la méthode		Ajouter Modifier Supprimer
Zone de texte	Zone de texte		

FIG. 5

25

20

Nom du paramètre		Champ texte
Description du paramètre	Valeurs possibles du paramètre	
Zone de texte	Zone de texte	
Type du paramètre	Champ texte	Valeur par défaut du paramètre
		Champ texte
Valeur réelle du paramètre		Champ texte

FIG.6

26

20

**THIS PAGE BLANK (USPTO)**



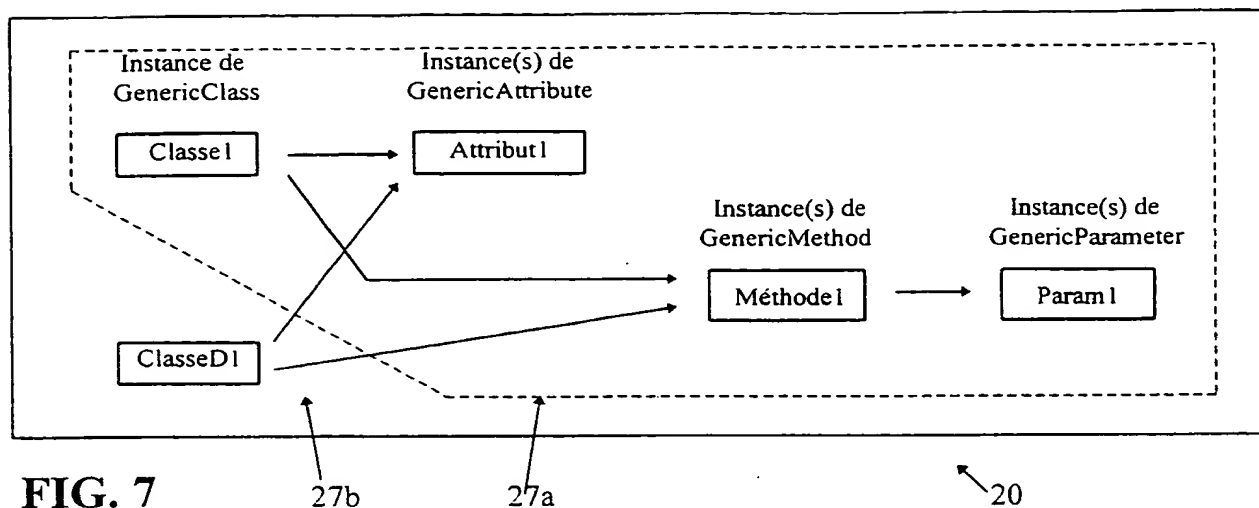


FIG. 7

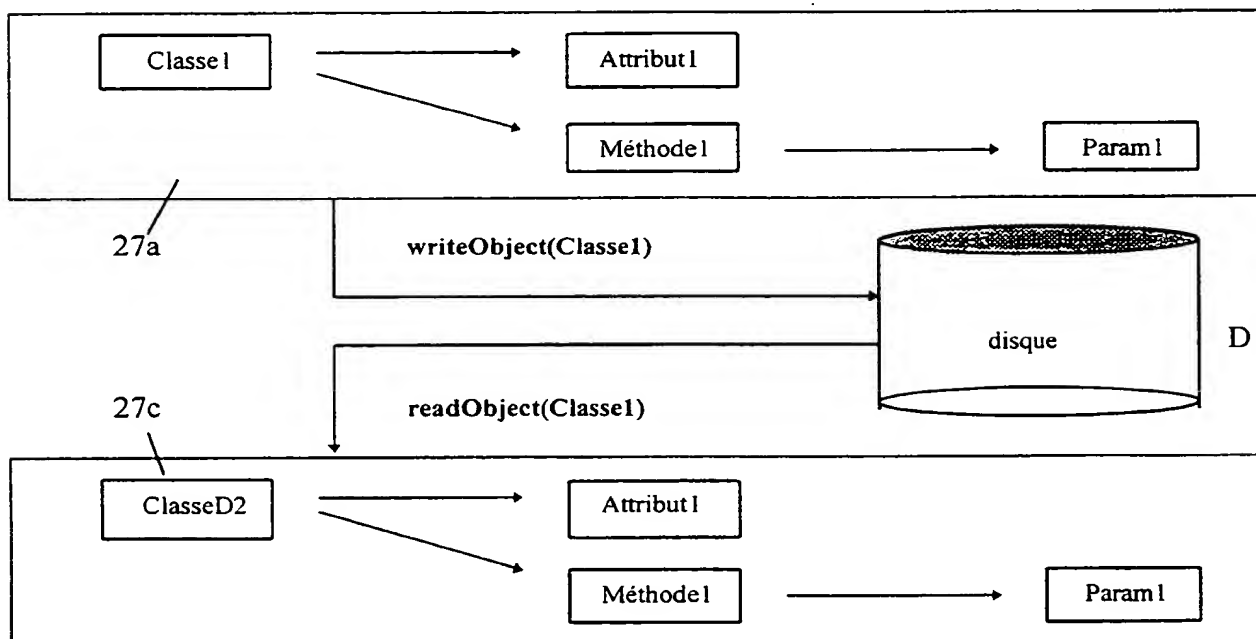


FIG. 8

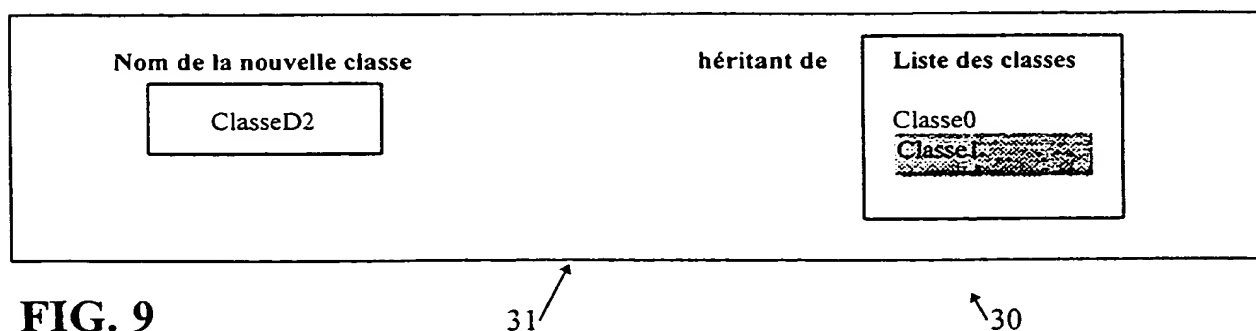


FIG. 9

**THIS PAGE BLANK (USPTO)**

Nom de la classe	ClasseD2	Identification de la classe	
Description de la classe		Parent de la classe	Classe1
Attributs de la classe	Attribut1	Modifier l'attribut	
		Supprimer l'attribut	
		Ajouter un attribut	
Méthodes de la classe	Méthode1	Modifier la méthode	
		Supprimer la méthode	
		Ajouter une méthode	
Classe	1		

FIG. 10

32

30

Liste de tous les étudiants ayant un âge inférieur à 26ans , étudiant à Paris et ayant fait un emprunt supérieur à 6000 francs .

FIG. 11

18a

19

À 10:30 le 02/05/1998	Exécution immédiate	Après exécution de la méthode "Méthode3" de l'objet "Objet6"
Impression de lettres à :		
<p>Liste de tous les étudiants ayant un âge inférieur à 26ans , étudiant à Paris et ayant fait un emprunt supérieur à 6000 francs .</p>		
Type de lettre : invitation d'anniversaire.		

FIG. 12

18b

41

19

**THIS PAGE BLANK (USPTO)**

Fichier	Documents	...	Politiques	Aide
	Ouvrir...		Afficher	
	Rechercher		Afficher l'historique	
	Enregistrer la classe		Enregistrer	
	Enregistrer la classe sous...		Enregistrer sous...	
	Fermer la classe		Fermer	

42

FIG. 13

Boolean
Character
Double
Float
Integer
Long
String
Void
List of Boolean
List of Character
List of Double
List of Float
List of Integer
List of Long
List of String

61

FIG. 14

**THIS PAGE BLANK (USPTO)**

# INTERNATIONAL SEARCH REPORT

Int'l. Application No.

PCT/FR 99/02629

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	OSAMU MIKAMI ET AL: "HETEROGENEOUS COMPUTING ENVIRONMENT "COUPLE" NEC RESEARCH AND DEVELOPMENT, vol. 32, no. 1, 1 January 1991 (1991-01-01), pages 130-141, XP000229428 ISSN: 0547-051X	1
A	abstract page 131, left-hand column, line 5 -page 133, left-hand column, line 12 page 135, right-hand column, line 5 -page 138, left-hand column, line 3; figures 5-8  -/-	2

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

1 March 2000

Date of mailing of the international search report

07/03/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Wiltink, J

# INTERNATIONAL SEARCH REPORT

International Application No

PCT/FR 99/02629

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	SHOWMAN P S: "AN OBJECT-BASED USER INTERFACE FOR THE HP NEWWAVE ENVIRONMENT" HEWLETT-PACKARD JOURNAL, vol. 40, no. 4, 1 August 1989 (1989-08-01), pages 9-17, XP000094705 page 11, left-hand column, line 44 -page 12, right-hand column, line 1	1
A	W0 96 37825 A (BULL HN INFORMATION SYST) 28 November 1996 (1996-11-28) abstract page 3, line 15 -page 5, line 2; claim 1	1,2



# INTERNATIONAL SEARCH REPORT

Information on patent family members

Int. Application No

PCT/FR 99/02629

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9637825 A	28-11-1996	US 5678047 A EP 0827606 A	14-10-1997 11-03-1998

**THIS PAGE BLANK (USPTO)**

# RAPPORT DE RECHERCHE INTERNATIONALE

Den. le Internationale No

PCT/FR 99/02629

## A. CLASSEMENT DE L'OBJET DE LA DEMANDE

CIB 7 G06F9/44

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

## B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)

CIB 7 G06F

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si réalisable, termes de recherche utilisés)

## C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie *	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
Y	OSAMU MIKAMI ET AL: "HETEROGENEOUS COMPUTING ENVIRONMENT "COUPLE"" NEC RESEARCH AND DEVELOPMENT, vol. 32, no. 1, 1 janvier 1991 (1991-01-01), pages 130-141, XP000229428 ISSN: 0547-051X	1
A	abrégé page 131, colonne de gauche, ligne 5 -page 133, colonne de gauche, ligne 12 page 135, colonne de droite, ligne 5 -page 138, colonne de gauche, ligne 3; figures 5-8	2



Voir la suite du cadre C pour la fin de la liste des documents



Les documents de familles de brevets sont indiqués en annexe

\* Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent

"E" document antérieur, mais publié à la date de dépôt international ou après cette date

"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention

"X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

"Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée

1 mars 2000

Date d'expédition du présent rapport de recherche internationale

07/03/2000

Nom et adresse postale de l'administration chargée de la recherche internationale

Office Européen des Brevets, P.B. 5818 Patentaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Fonctionnaire autorisé

Wiltink, J

## C.(suite) DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
Y	SHOWMAN P S: "AN OBJECT-BASED USER INTERFACE FOR THE HP NEWWAVE ENVIRONMENT" HEWLETT-PACKARD JOURNAL, vol. 40, no. 4, 1 août 1989 (1989-08-01), pages 9-17, XP000094705 page 11, colonne de gauche, ligne 44 -page 12, colonne de droite, ligne 1	1
A	W0 96 37825 A (BULL HN INFORMATION SYST) 28 novembre 1996 (1996-11-28) abrégé page 3, ligne 15 -page 5, ligne 2; revendication 1	1,2

# RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Den. Je internationale No

PCT/FR 99/02629

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
WO 9637825 A	28-11-1996	US 5678047 A EP 0827606 A	14-10-1997 11-03-1998

**THIS PAGE BLANK (USPTO)**